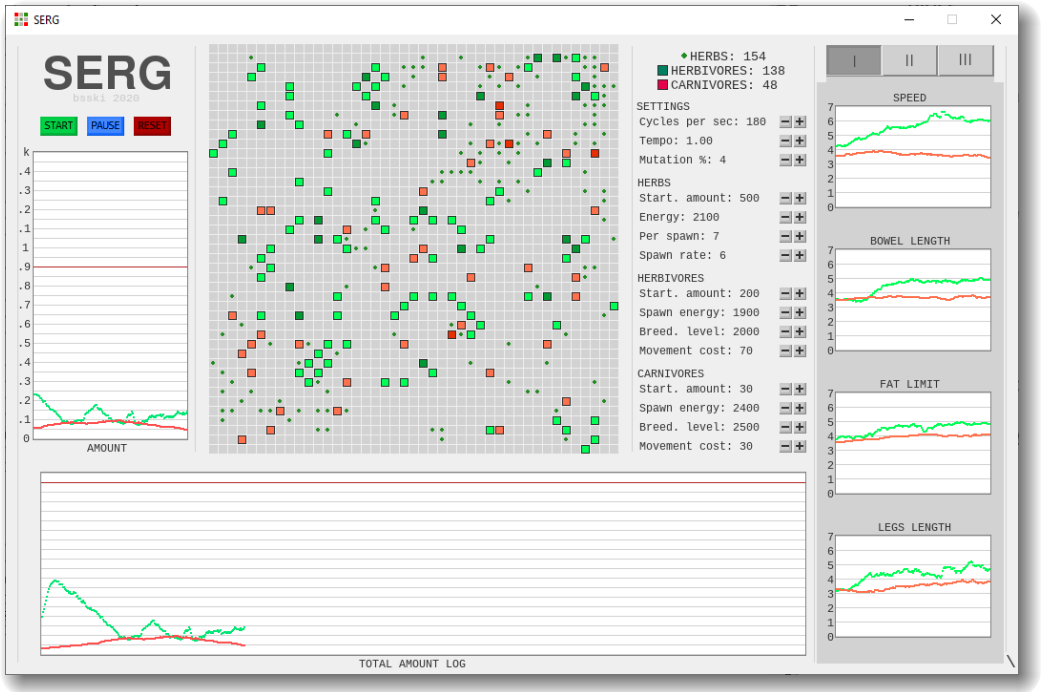
Przestrzenna symulacja iterowanego dylematu więźnia rozgrywanego między sztucznymi sieciami neuronowymi w paradygmacie uczenia ze wzmocnieniem

Bartosz Swędrowski

23 kwietnia 2021

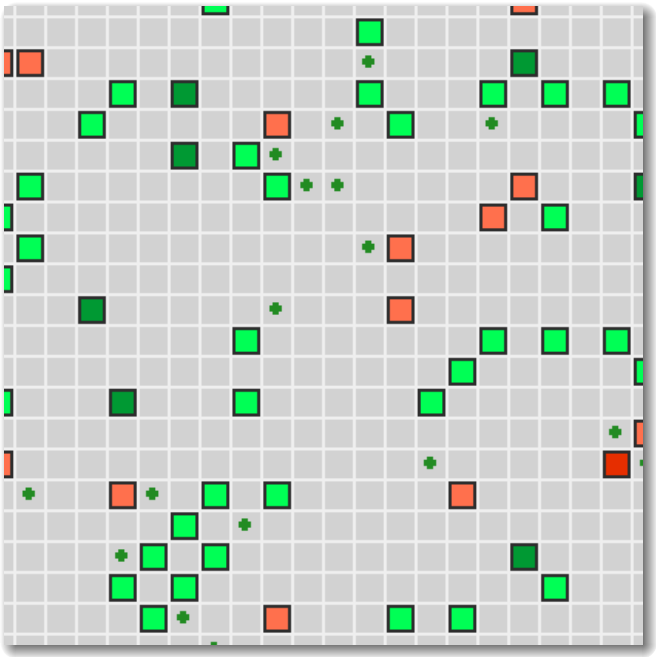
Plan prezentacji

SERG



Projekt dodatkowy z przedmiotu Sztuczna inteligencja - SERG (Symulator Ewolucyjnej Rywalizacji Genów), realizowany pod opieką dr. Andrzeja Gajdy.

Dlaczego nie SERG?



Wycinek planszy SERGa

- **plansza**
- **agenci z parametrami** i rolami
- potencjał na licencjat
- niestety brak miejsca na **sieci neuronowe**

Przykładowa tabela nagród w dylemacie więźnia

Gracz1	Gracz2	Nagroda [G1, G2]
0	0	[3, 3]
0	1	[-1, 4]
1	0	[4, -1]
1	1	[0, 0]

0 - próba współpracy
1 - próba oszustwa

Schemat interakcji między agentami (jeden epizod)

Gracz1	0
Gracz2	0

Schemat interakcji między agentami (jeden epizod)

Gracz1	0	1
Gracz2	0	0

Schemat interakcji między agentami (jeden epizod)

Gracz1	0	1	1
Gracz2	0	0	1

Schemat interakcji między agentami (jeden epizod)

Gracz1	0	1	1	0
Gracz2	0	0	1	1

Schemat interakcji między agentami (jeden epizod)

Gracz1	0	1	1	0	1	1	1	1	1	1
Gracz2	0	0	1	1	1	1	1	1	1	1

Przykładowe taktyki

1. Always Cooperate
2. Always Defect
3. Tit For Tat
4. GRIM
5. Random

Co składa się na HIVE?

1. plansza
2. agenci w grupach (hives - roje)
3. interagują na zasadach iterowanego dylematu więźnia
4. otrzymują decyzje o akcjach od sieci neuronowej ich grupy
5. ale przepuszczają ją przez „filtr” swojego parametru

Paradygmaty uczenia maszynowego

- uczenie nadzorowane (supervised learning)

Paradygmaty uczenia maszynowego

- uczenie nadzorowane (supervised learning)
- uczenie nienadzorowane (unsupervised learning)

Paradygmaty uczenia maszynowego

- uczenie nadzorowane (supervised learning)
- uczenie nienadzorowane (unsupervised learning)
- **uczenie ze wzmocnieniem (reinforcement learning)**

Paradygmaty uczenia maszynowego

- uczenie nadzorowane (supervised learning)
- uczenie nienadzorowane (unsupervised learning)
- **uczenie ze wzmocnieniem (reinforcement learning)**
  - głębokie uczenie ze wzmocnieniem (deep reinforcement learning)

Paradygmaty uczenia maszynowego

- uczenie nadzorowane (supervised learning)
- uczenie nienadzorowane (unsupervised learning)
- **uczenie ze wzmocnieniem (reinforcement learning)**
  - głębokie uczenie ze wzmocnieniem (deep reinforcement learning)
    - algorytm DQN (Deep Q-Network) w rozszerzonym wariancie DDDQN (Dueling Double Deep Q-Network)

Q-learning

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
s_0	490	490	490	700	490	490	490	490
s_1	490	490	700	490	490	490	490	490
s_2	490	490	490	700	700	490	490	490
s_3	490	490	700	1000	700	700	490	490
s_4	490	490	490	700	700	700	490	490
s_5	490	490	700	700	1000	700	700	490
s_6	490	490	490	490	700	700	490	490
s_7	490	490	490	490	700	1000	700	490
s_8	700	490	490	490	700	700	1000	700
s_9	700	490	490	490	490	700	700	1000
s_{10}	700	490	490	490	490	490	490	490
s_{11}	490	490	490	490	490	700	490	490
s_{12}	490	490	490	490	490	700	700	490
s_{13}	490	490	490	490	490	700	700	700
s_{14}	490	490	490	490	490	490	700	700
s_{15}	490	490	490	490	490	490	490	700

Rysunek: Q-table

Sposób obliczania nowego q-value

$$\text{New } Q(s,a) = Q(s,a) + \alpha\,[R(s,a) + \gamma \max Q'(s',a') - Q(s,a)]$$

- New Q Value for that state and the action
- Learning Rate
- Reward for taking that action at that state
- Current Q Values
- Maximum expected future reward given the new state (s') and all possible actions at that new state.
- Discount Rate

DQN

1. DQN (Deep Q-network)

DQN

1. DQN (Deep Q-network)
2. Double DQN - DQN w podstawowej wersji ma tendencję do przeszacowywania wartości stanów. W Double DQN online network generuje akcje, podczas gdy target network estymuje Q value.

DQN

1. DQN (Deep Q-network)
2. Double DQN - DQN w podstawowej wersji ma tendencję do przeszacowywania wartości stanów. W Double DQN online network generuje akcje, podczas gdy target network estymuje Q value.
3. Dueling DQN - Dueling DQN rozdziela sieć w przedostatniej warstwie na dwie części: jedną estymującą *q-value* i jedną generującą akcję. Następnie znów łączy sieć w całość. Podobnie jak wyżej, powodem jest to, że DQN jest "nazbyt optymistyczne" jeśli chodzi o estymację q-values.

DQN

1. DQN (Deep Q-network)
2. Double DQN - DQN w podstawowej wersji ma tendencję do przeszacowywania wartości stanów. W Double DQN online network generuje akcje, podczas gdy target network estymuje Q value.
3. Dueling DQN - Dueling DQN rozdziela sieć w przedostatniej warstwie na dwie części: jedną estymującą *q-value* i jedną generującą akcję. Następnie znów łączy sieć w całość. Podobnie jak wyżej, powodem jest to, że DQN jest "nazbyt optymistyczne" jeśli chodzi o estymację q-values.
4. Double Dueling DQN (DDDQN) - połączenie obu wariantów wymienionych wyżej.

Właściwości algorytmu DQN i HIVE

1. Jest to uczenie ze wzmocnieniem typu *model-free* - środowisko jest dla agenta czarną skrzynką. Nie ma on do niego dostępu, a wiedza jaką o nim posiądzie pochodzi z działania metodą prób i błędów.

Właściwości algorytmu DQN i HIVE

1. Jest to uczenie ze wzmocnieniem typu *model-free* - środowisko jest dla agenta czarną skrzynką. Nie ma on do niego dostępu, a wiedza jaką o nim posiądzie pochodzi z działania metodą prób i błędów.
2. dyskretna przestrzeń akcji - liczba możliwych do wykonania akcji jest dyskretna i skończona,

Właściwości algorytmu DQN i HIVE

1. Jest to uczenie ze wzmocnieniem typu *model-free* - środowisko jest dla agenta czarną skrzynką. Nie ma on do niego dostępu, a wiedza jaką o nim posiądzie pochodzi z działania metodą prób i błędów.
2. dyskretna przestrzeń akcji - liczba możliwych do wykonania akcji jest dyskretna i skończona,
3. polityka typu *epsilon greedy* - stochastyczna polityka działania agenta - mapuje ona stany na rozkład prawdopodobieństwa,

Właściwości algorytmu DQN i HIVE

1. Jest to uczenie ze wzmocnieniem typu *model-free* - środowisko jest dla agenta czarną skrzynką. Nie ma on do niego dostępu, a wiedza jaką o nim posiądzie pochodzi z działania metodą prób i błędów.
2. dyskretna przestrzeń akcji - liczba możliwych do wykonania akcji jest dyskretna i skończona,
3. polityka typu *epsilon greedy* - stochastyczna polityka działania agenta - mapuje ona stany na rozkład prawdopodobieństwa,
4. metoda *off-policy* - algorytm ocenia i ulepsza inną politykę, niż ta, która jest wykorzystywana do generowania akcji,

Właściwości algorytmu DQN i HIVE

1. Jest to uczenie ze wzmocnieniem typu *model-free* - środowisko jest dla agenta czarną skrzynką. Nie ma on do niego dostępu, a wiedza jaką o nim posiądzie pochodzi z działania metodą prób i błędów.
2. dyskretna przestrzeń akcji - liczba możliwych do wykonania akcji jest dyskretna i skończona,
3. polityka typu *epsilon greedy* - stochastyczna polityka działania agenta - mapuje ona stany na rozkład prawdopodobieństwa,
4. metoda *off-policy* - algorytm ocenia i ulepsza inną politykę, niż ta, która jest wykorzystywana do generowania akcji,
5. częściowo obserwowalne środowisko - algorytm nie ma dostępu do kompletnej wiedzy o stanie środowiska, w danym momencie jest dla niego dostępna tylko jego część.

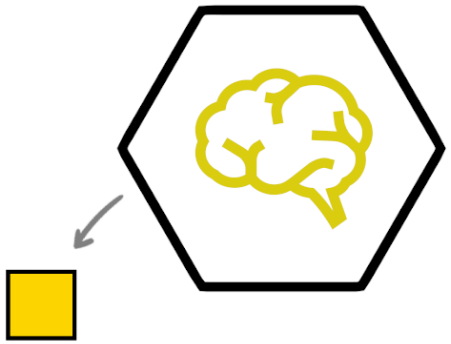
Czym więc jest HIVE?

HIVE

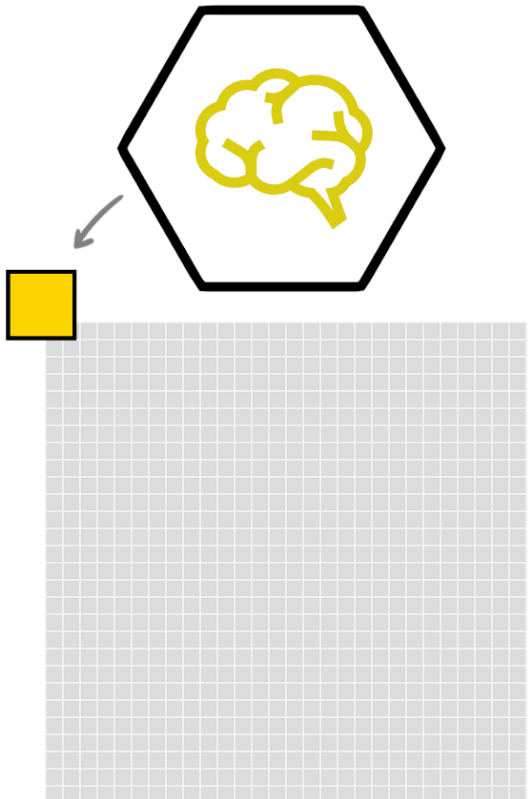
Model środowiska, w którym agenci o częściowo ograniczonej racjonalności funkcjonują w grupach i wchodzą ze sobą w interakcję bazowaną na iterowanym dylemacie więźnia, podejmując decyzję na podstawie obliczeń sieci neuronowej ich grupy zmodyfikowanych przez wartość parametru poszczególnych agentów.

Ilustracja idei

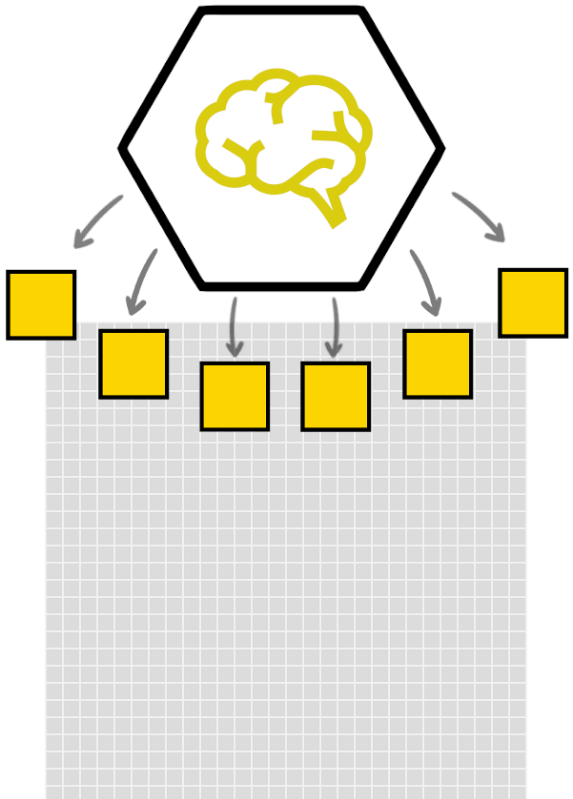
Ilustracja idei



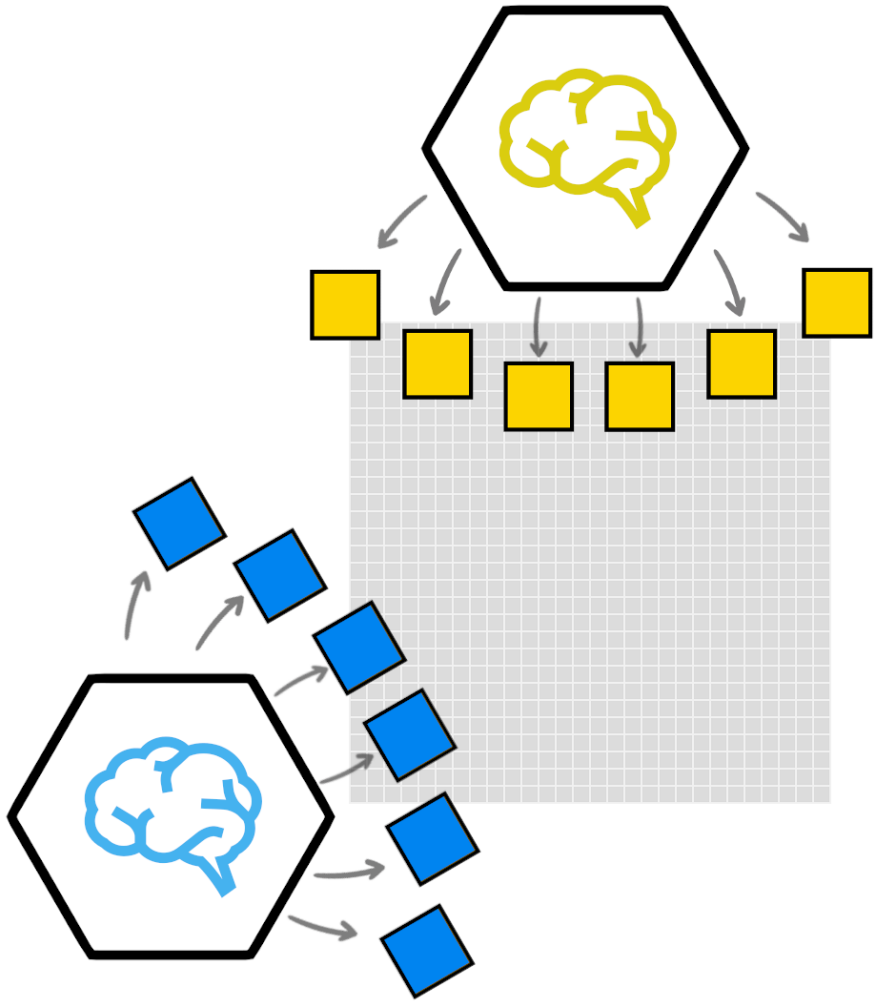
Ilustracja idei



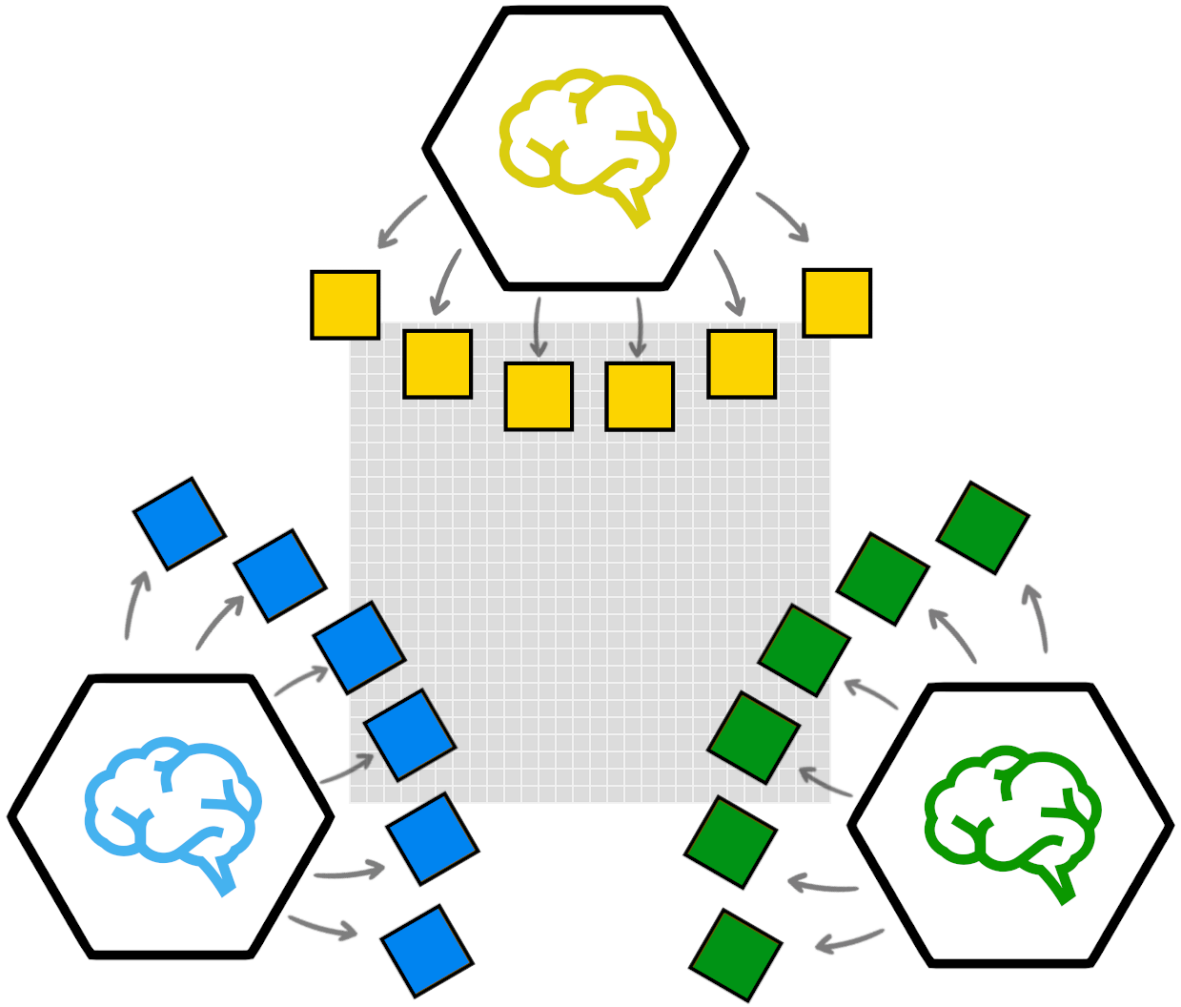
Ilustracja idei



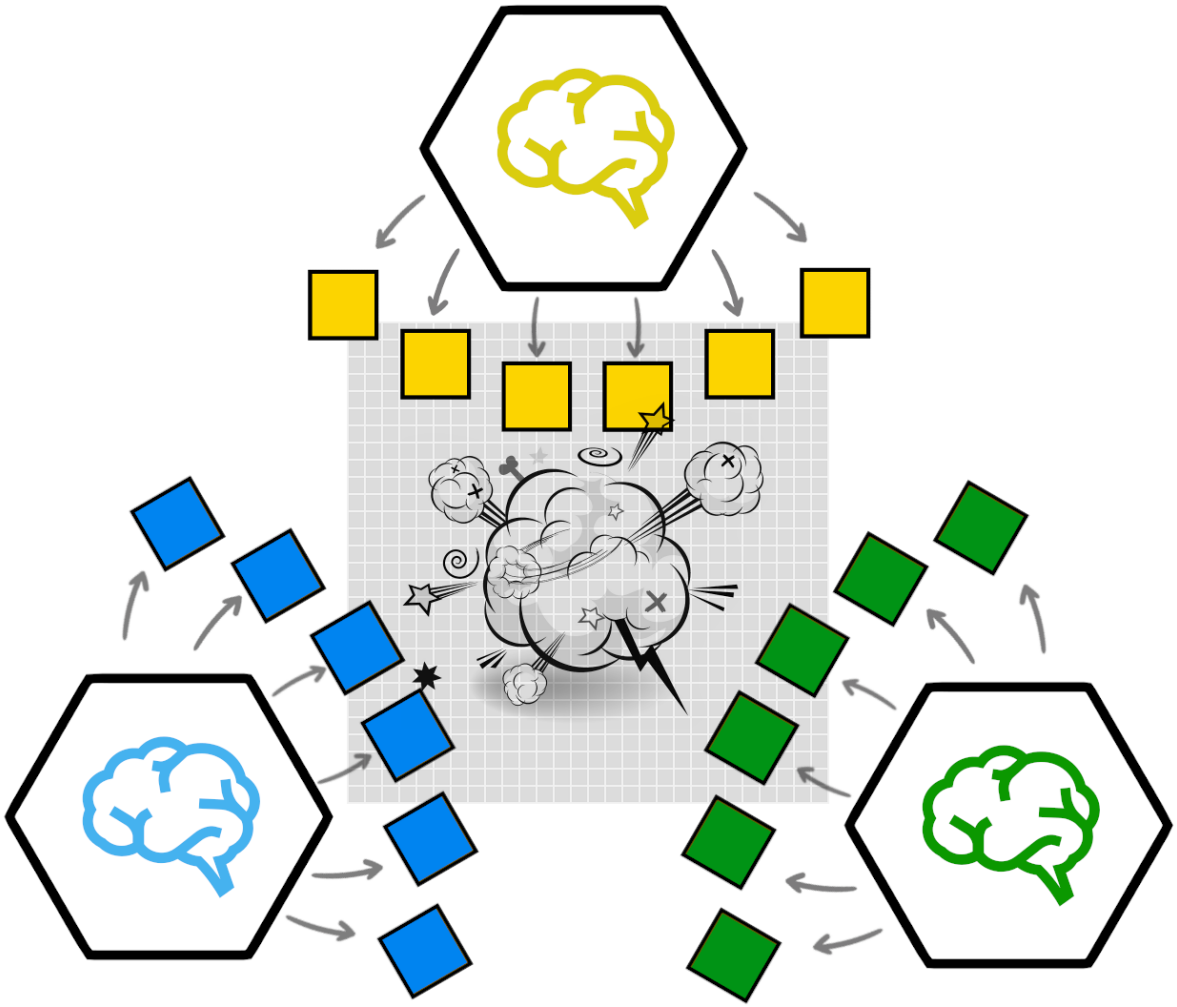
Ilustracja idei



Ilustracja idei



Ilustracja idei

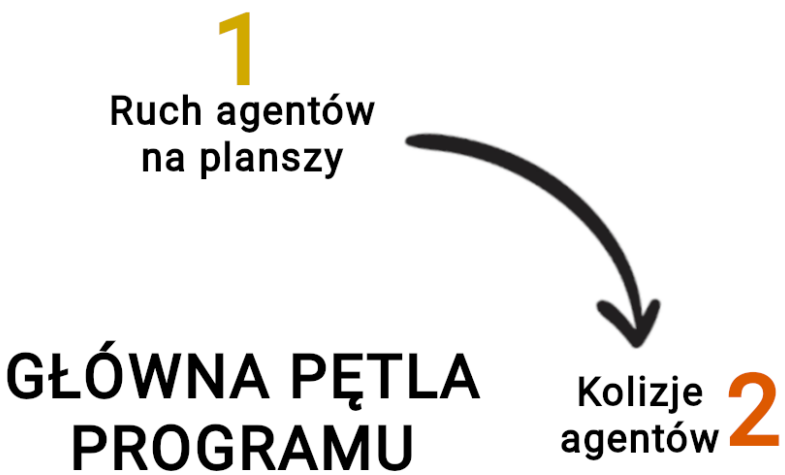


Główna pętla programu

1
Ruch agentów na planszy

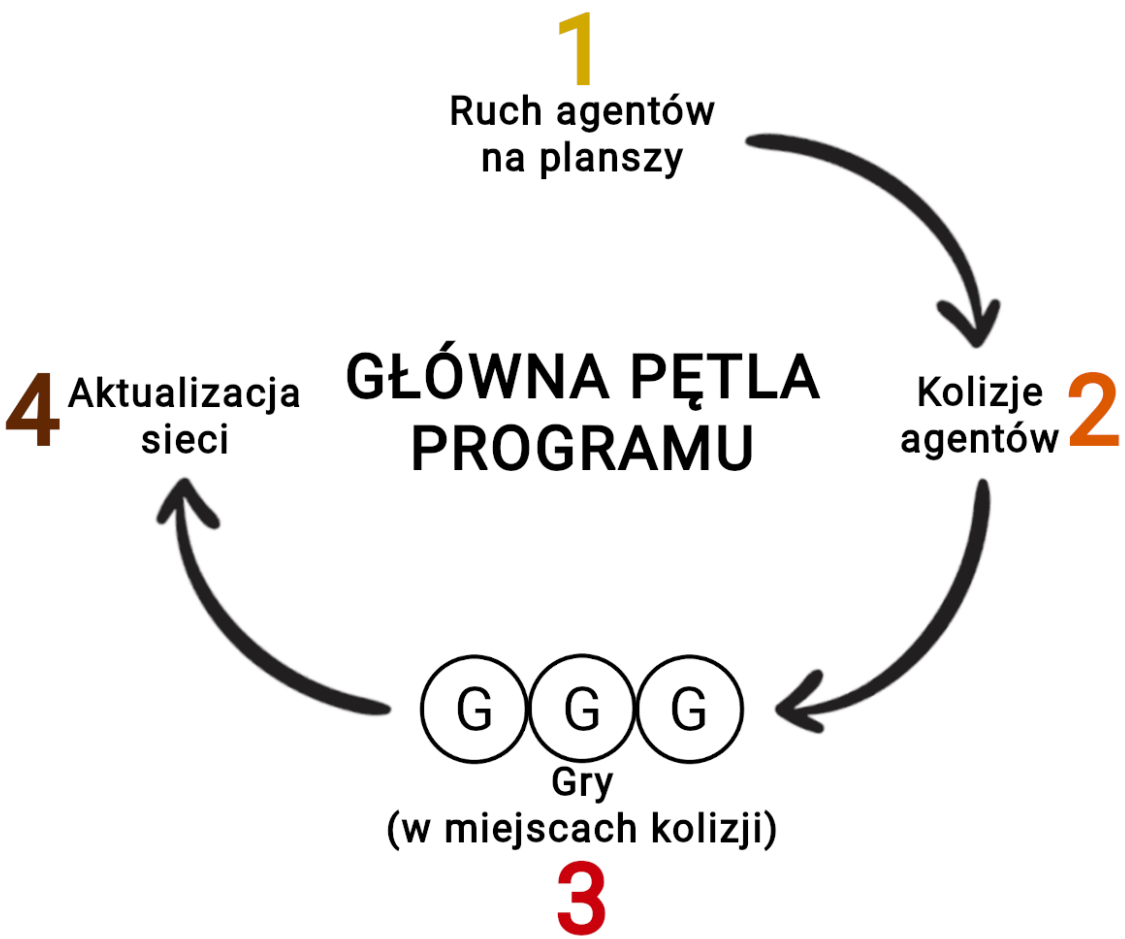
GŁÓWNA PĘTLA PROGRAMU

Główna pętla programu

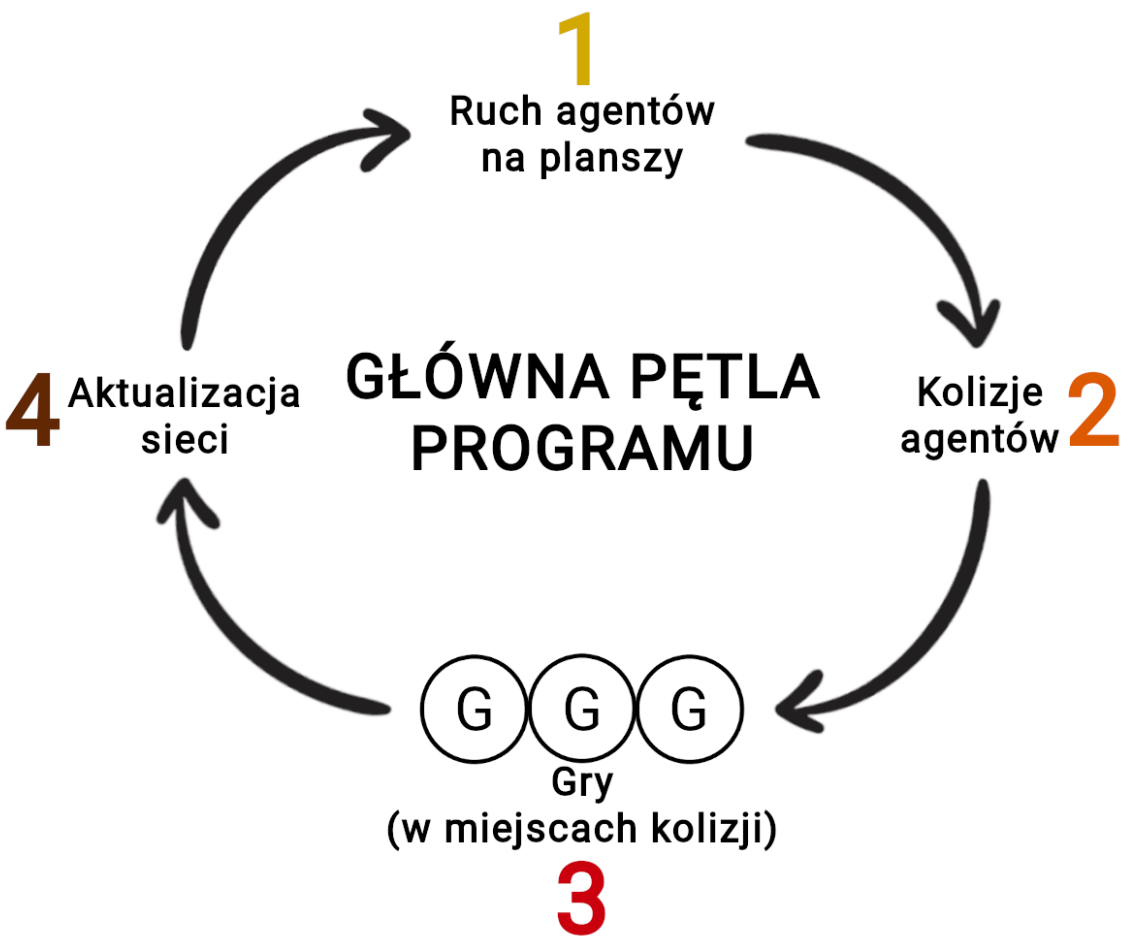


Główna pętla programu

Główna pętla programu



Główna pętla programu



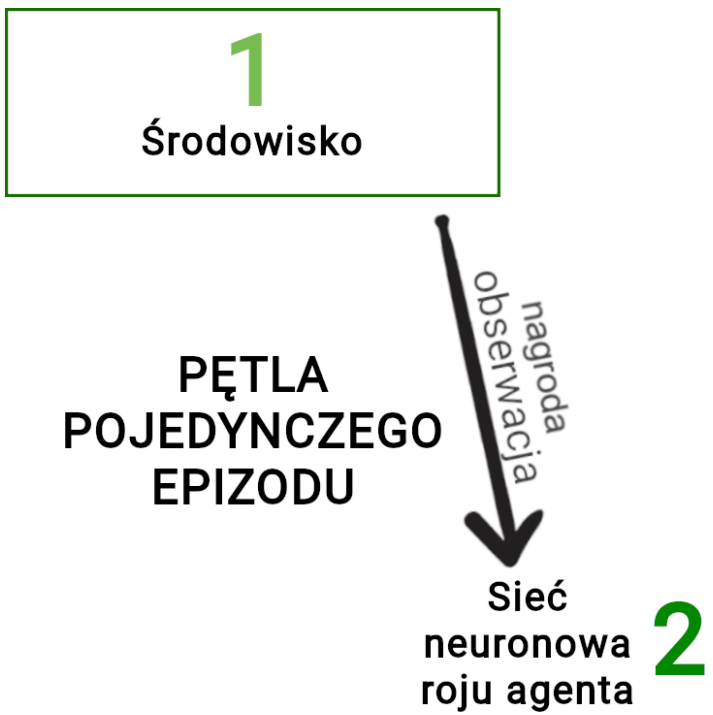
Pętla epizodu z perspektywy jednego z graczy

1
Środowisko

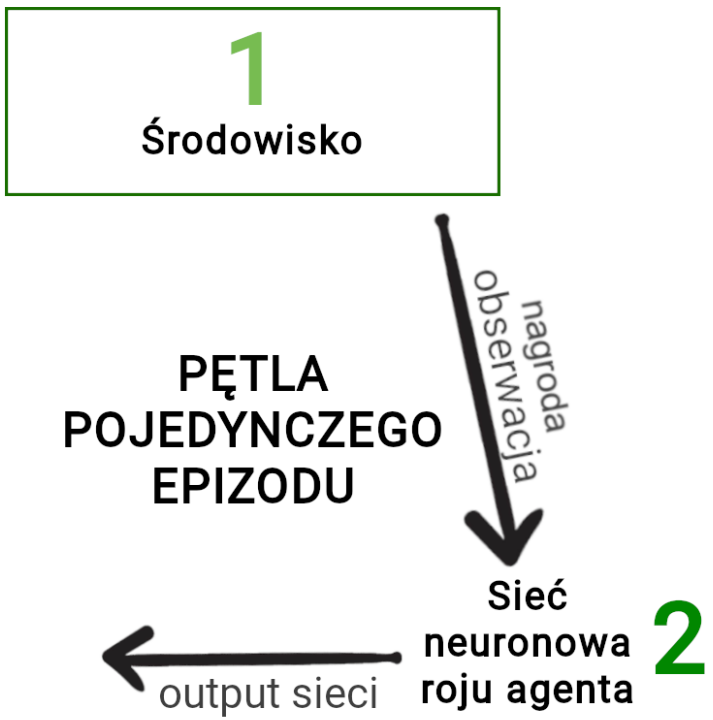
PĘTLA
POJEDYNCZEGO
EPIZODU

Pętla epizodu z perspektywy jednego z graczy

Pętla epizodu z perspektywy jednego z graczy



Pętla epizodu z perspektywy jednego z graczy



Pętla epizodu z perspektywy jednego z graczy

Pętla epizodu z perspektywy jednego z graczy

Na jakiej podstawie sieć generuje akcję? Input sieci

1. numer aktualnej wymiany (1-20)
2. własna akcja z poprzedniej wymiany
3. akcja przeciwnika z poprzedniej wymiany
4. własny numer roju
5. numer roju przeciwnika
6. moja nagroda z poprzedniej wymiany
7. własny indeks agenta
8. indeks agenta przeciwnika
9. + te same informacje dla poprzednich dwóch wymian

Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,

Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,
2. optimizer typu Adam (Adaptive Moment Estimation),

Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,
2. optimizer typu Adam (Adaptive Moment Estimation),
3. współczynnik uczenia = 1e-3,

Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,
2. optimizer typu Adam (Adaptive Moment Estimation),
3. współczynnik uczenia = 1e-3,
4. gamma = 0.9,

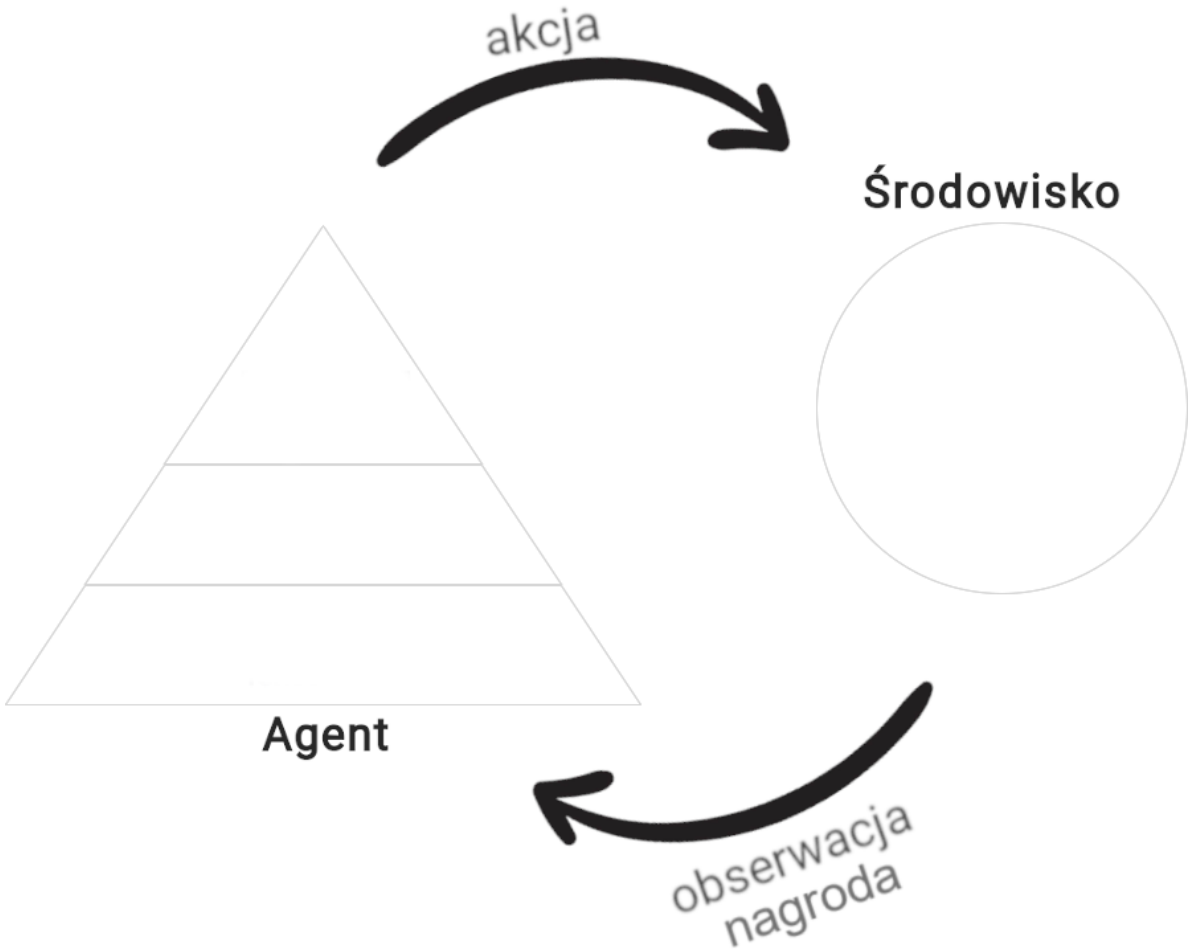
Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,
2. optimizer typu Adam (Adaptive Moment Estimation),
3. współczynnik uczenia = 1e-3,
4. gamma = 0.9,
5. wielkość "okna" obserwacji (window length) = 3,

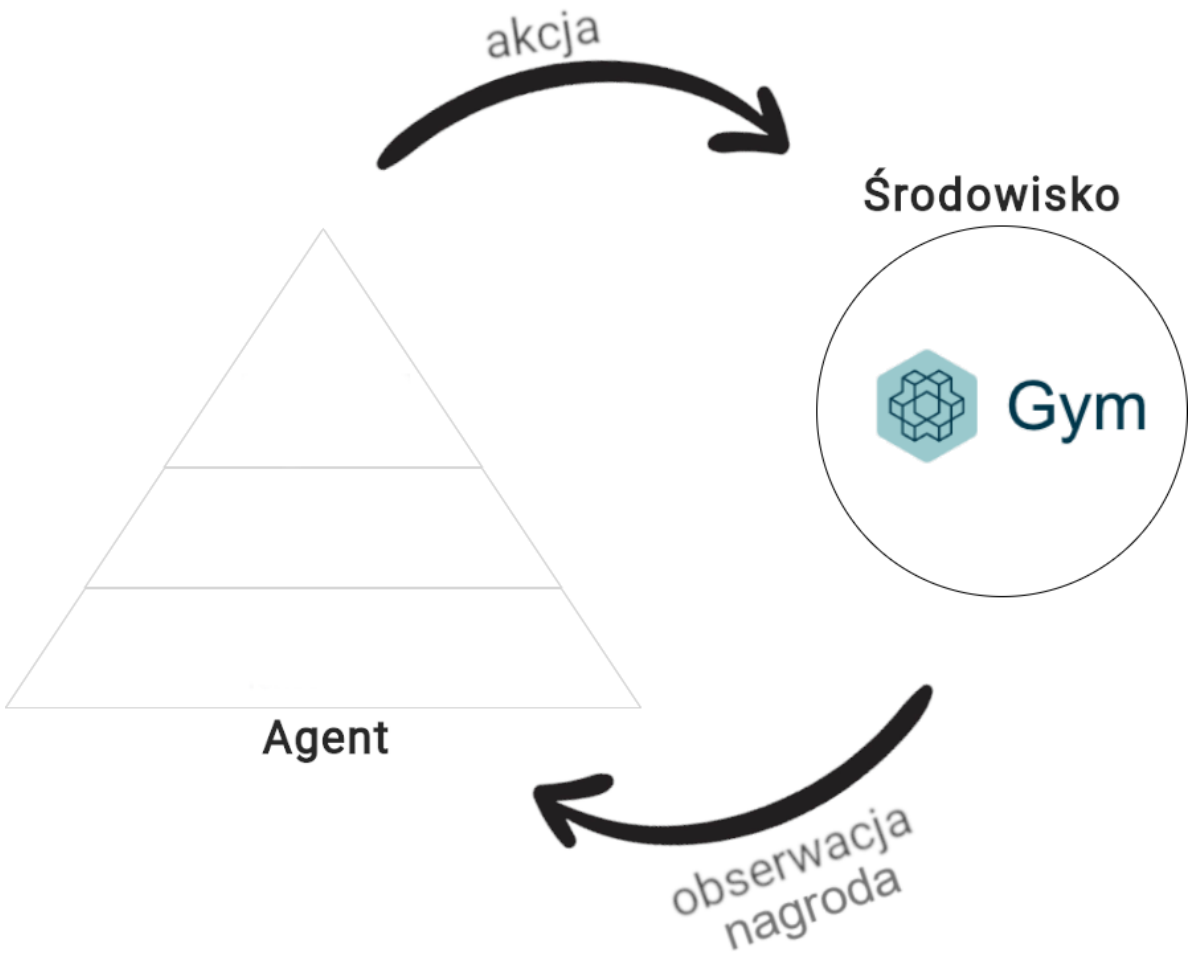
Hiperparametry

1. Sequential memory o pojemności z przedziału 2000-32000,
2. optimizer typu Adam (Adaptive Moment Estimation),
3. współczynnik uczenia = 1e-3,
4. gamma = 0.9,
5. wielkość "okna" obserwacji (window length) = 3,
6. architektura sieci: (24, input), (128, ReLu), (96, ReLu), (64, ReLu), (32, ReLu), (16, ReLu), (2, Sigmoid) lub (10, Sigmoid).

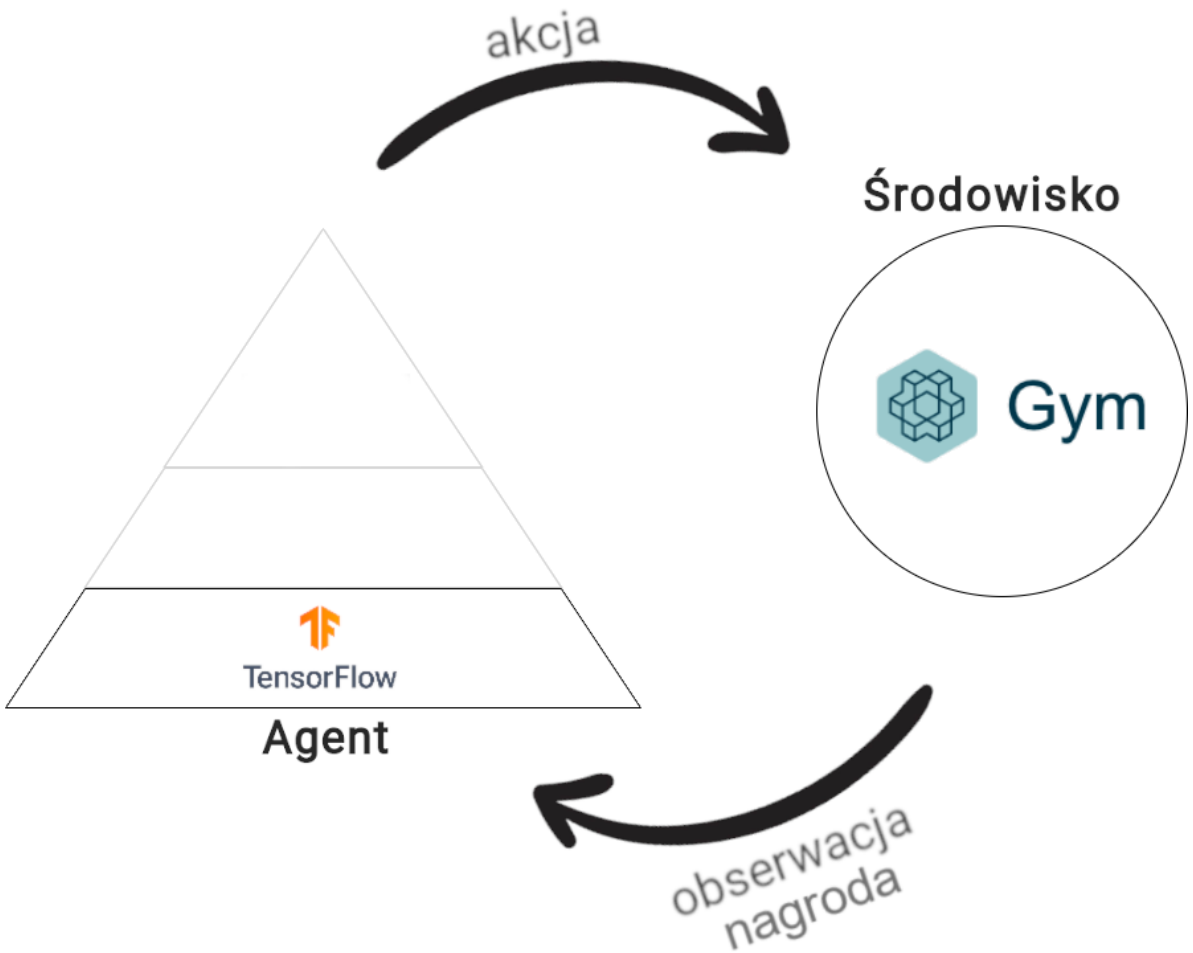
Tech stack: struktura programu



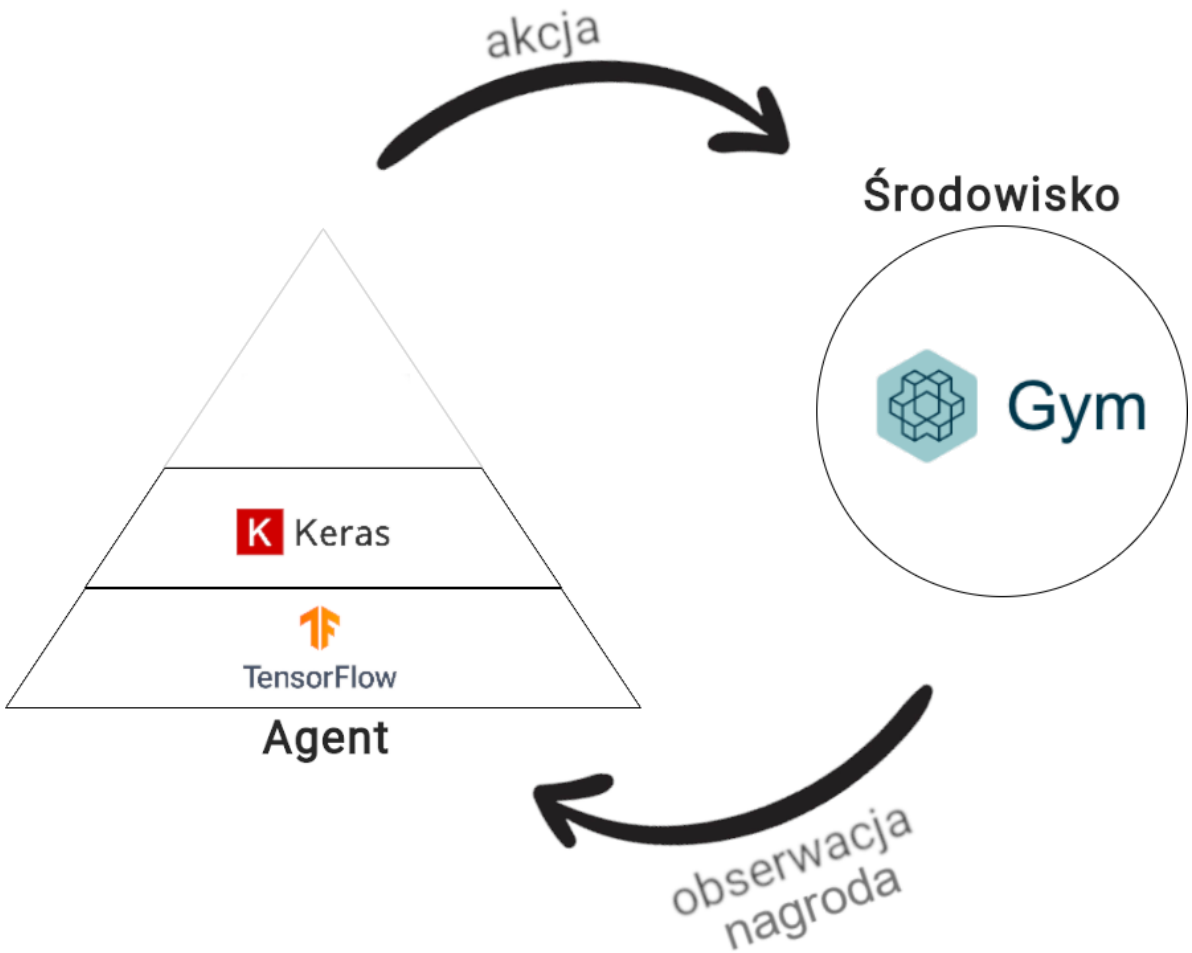
Tech stack: struktura programu



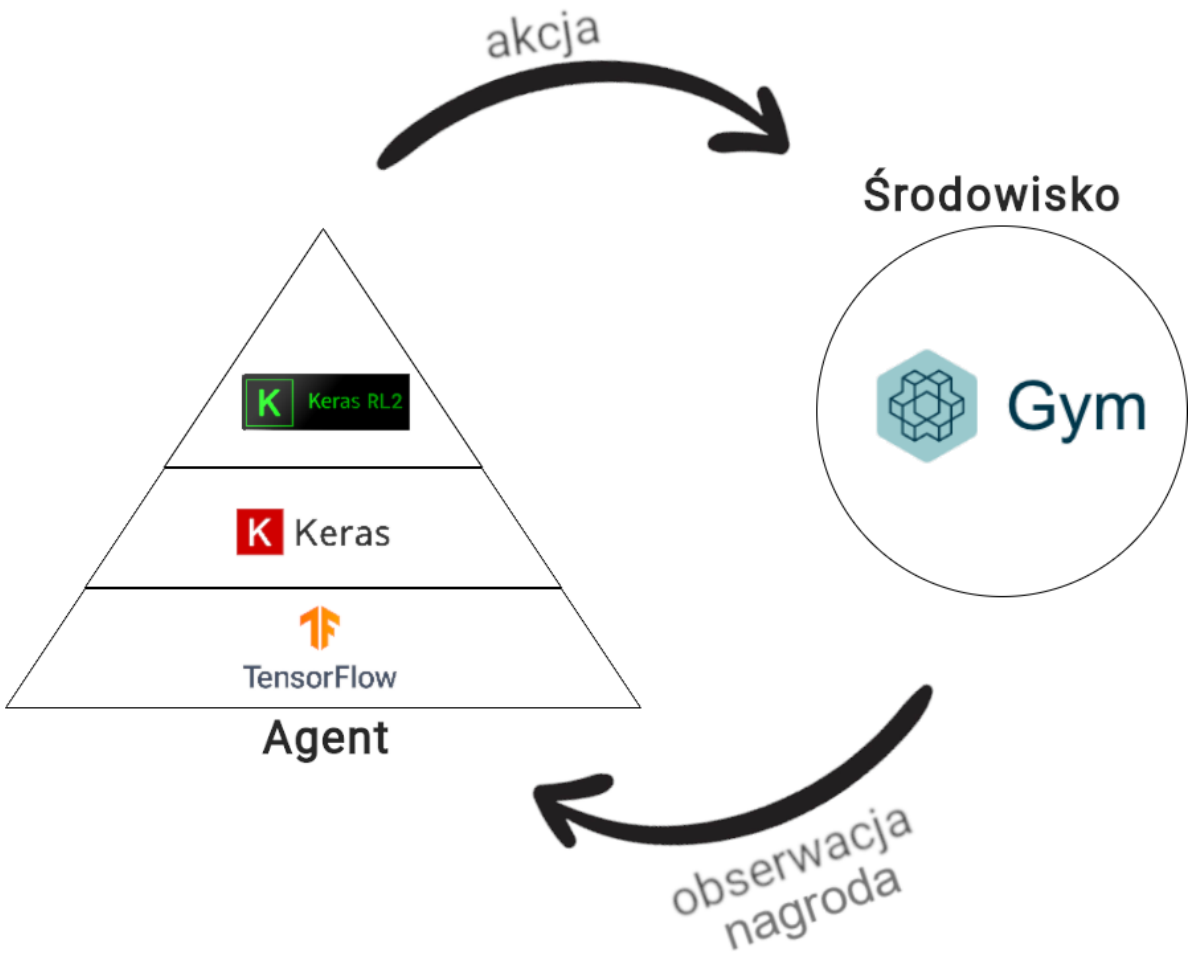
Tech stack: struktura programu



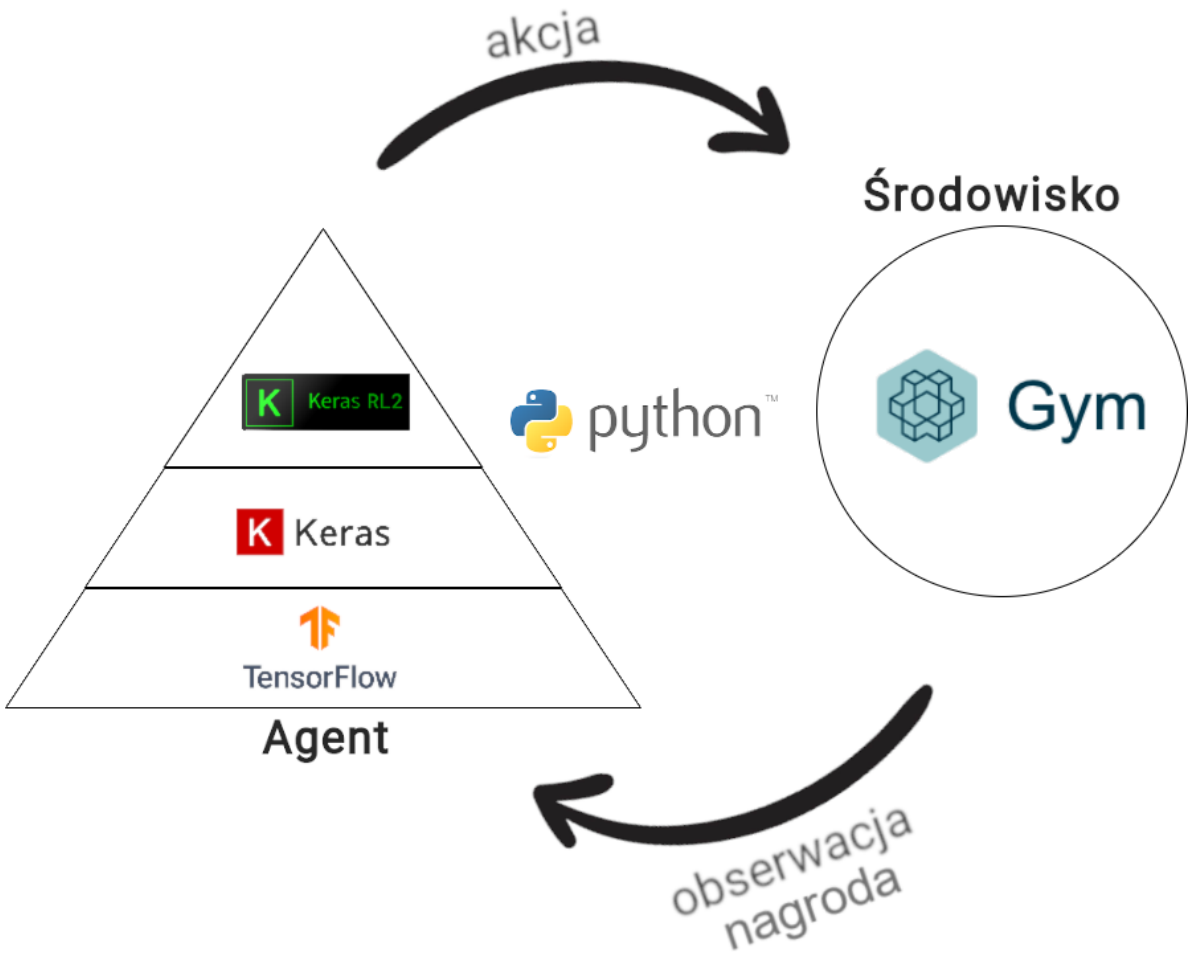
Tech stack: struktura programu



Tech stack: struktura programu



Tech stack: struktura programu



Zaplanowane testy

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}
3. Sześć sieci, rozgrywki między sobą.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}
3. Sześć sieci, rozgrywki między sobą.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}
4. Sześć sieci, rozgrywki między sobą + parametr.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}
3. Sześć sieci, rozgrywki między sobą.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}
4. Sześć sieci, rozgrywki między sobą + parametr.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}

Zaplanowane testy

1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}
3. Sześć sieci, rozgrywki między sobą.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}
4. Sześć sieci, rozgrywki między sobą + parametr.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}

Czynnikiem różnicującym sieci w testach jest pojemność pamięci.

Zaplanowane testy

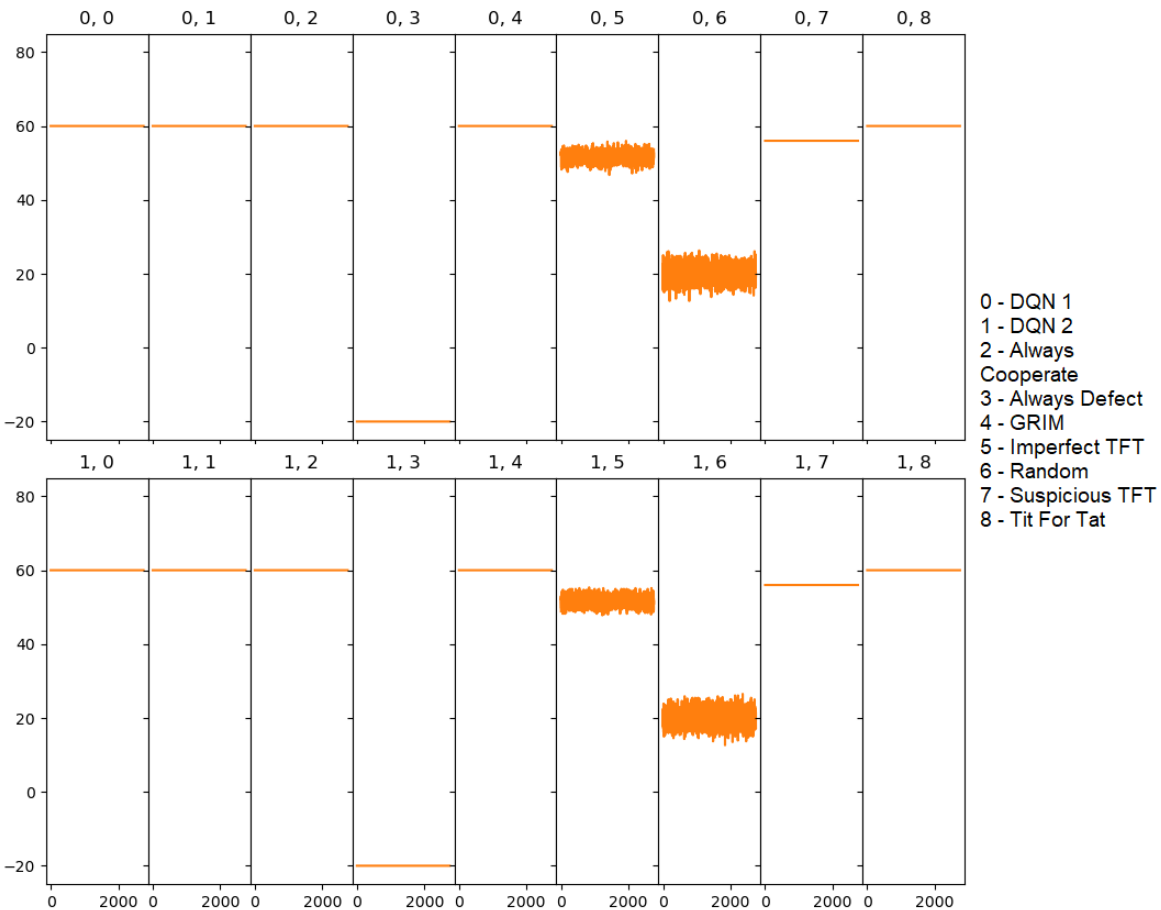
1. Dwie sieci vs stałe taktyki.
   Pamięci sieci: {1: 8000, 2: 32000}
2. Dwie sieci vs stałe taktyki + parametr.
   Pamięci sieci: {1: 8000, 2: 32000}
3. Sześć sieci, rozgrywki między sobą.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}
4. Sześć sieci, rozgrywki między sobą + parametr.
   Pamięci sieci: {1: 2000, 2: 8000, 3: 14000, 4: 20000, 5: 26000, 6: 32000}

Czynnikiem różnicującym sieci w testach jest pojemność pamięci. Zakres parametru: [-3, 3]; wpływ parametru: (akcja + 9*(0.16 * parametr))

Wstępne wyniki 1

1. Sieci efektywnie grają przeciwko taktykom Always Cooperate, GRIM, Tit For Tat, Suspicious TFT, Imperfect TFT oraz Random, osiągając optymalne wyniki. Sieci przegrywają w rozgrywkach przeciwko taktyce Always Defect. Można podejrzewać faworyzowanie akcji współpracy - najprawdopodobniej rozwiązaniem jest zbalansowanie tabeli nagród w stronę większych korzyści z udanego oszustwa.

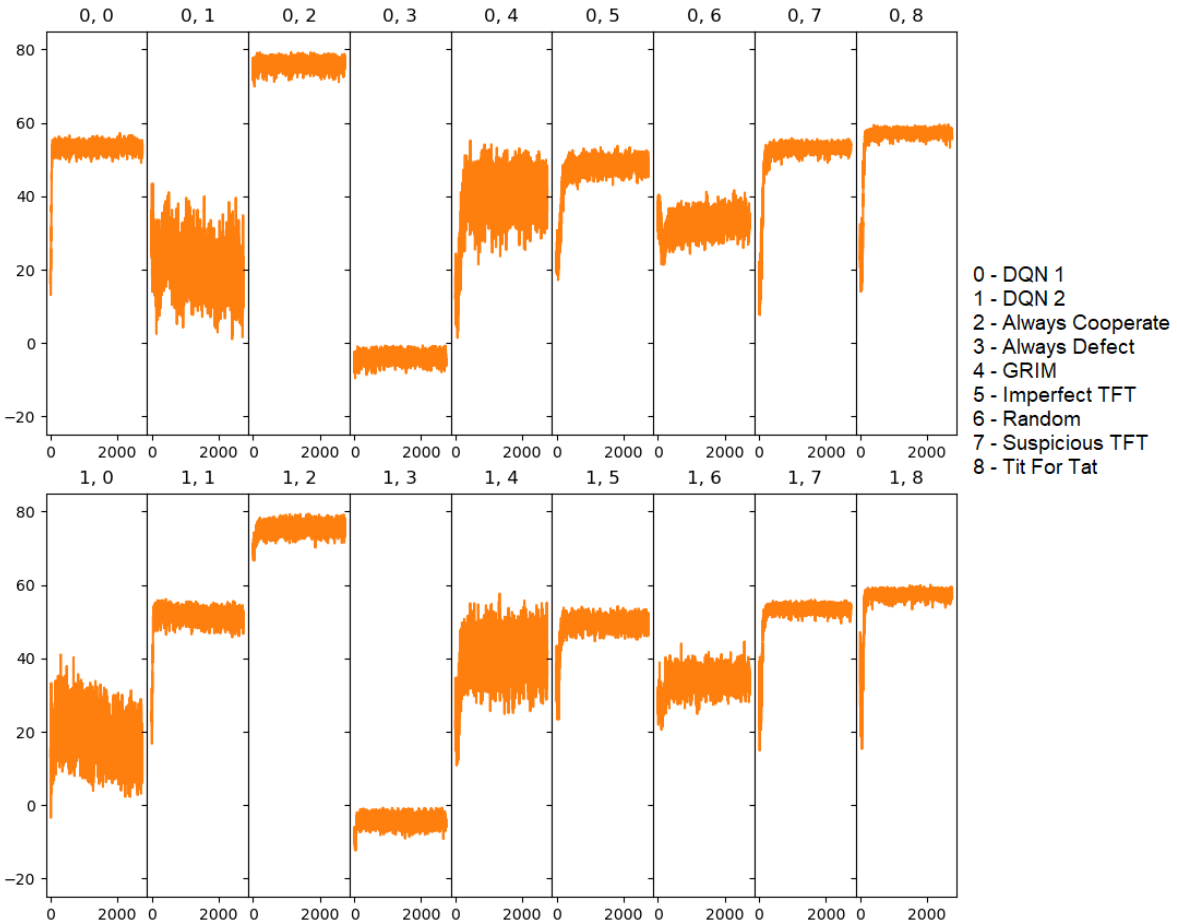
Wstępne wyniki 1



Wstępne wyniki 2

1. Sieci efektywnie grają przeciwko taktykom Always Cooperate, GRIM, Tit For Tat, Suspicious TFT, Imperfect TFT oraz Random, osiągając optymalne wyniki. Sieci przegrywają w rozgrywkach przeciwko taktyce Always Defect. Można podejrzewać faworyzowanie akcji współpracy - najprawdopodobniej rozwiązaniem jest zbalansowanie tabeli nagród w stronę większych korzyści z udanego oszustwa.

Wstępne wyniki 2



Wstępne wyniki 3

Pełna współpraca między agentami - średnia nagroda 60 w każdej interakcji.

Wstępne wyniki 4

Wykresy zbyt mało oczywiste do analizy bez testów statystycznych. Potrzeba większej ilości danych.

Planowane analizy i przyszłe prace

W planie: wykonanie finalnych obliczeń dla wyżej wymienionych eksperymentów oraz zbadanie ich jednoczynnikową analizą wariancji z powtarzanym pomiarem.
Przyszłe prace: zakodować cechy jakościowe danych wejściowych wg formatu "one-hot encoding".

Podsumowanie

HIVE

Przestrzenna symulacja iterowanego dylematu więźnia rozgrywanego między sztucznymi sieciami neuronowymi w paradygmacie uczenia ze wzmocnieniem.

Uznanie

- Wektory strzałek: https://www.freepik.com/Harryarts
- obrazek mózgu: Creative Commons