Czy na MuZero można odpalić Doom'a?

Paulina Brzęcka Marek Borzyszkowski

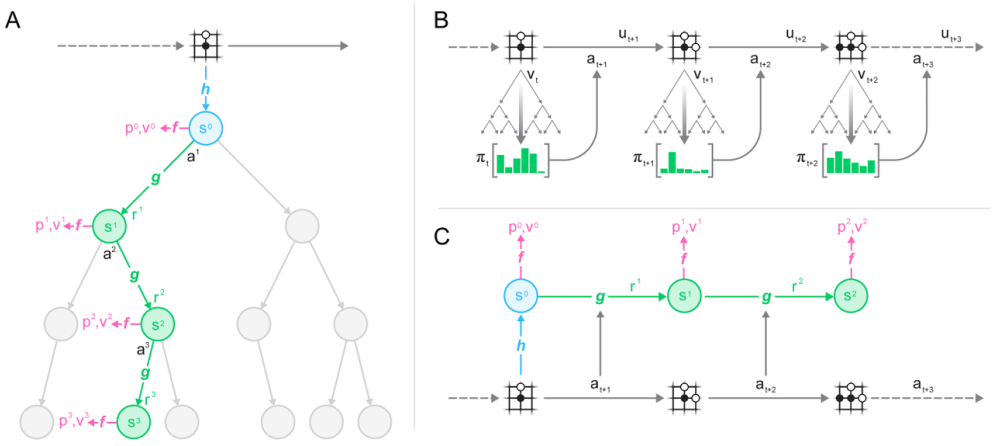
23 stycznia 2025

Spis treści

1 Jak działa MuZero

MuZero działa w oparciu o zaawansowany model uczenia maszynowego, który łączy w sobie planowanie, uczenie się oraz symulację, aby optymalizować podejmowanie decyzji w różnych środowiskach. Kluczowe cechy działania MuZero to:

- Modelowanie wewnętrzne - W przeciwieństwie do wcześniejszych algorytmów (np. AlphaGo czy AlphaZero), MuZero nie wymaga znajomości zasad gry ani pełnego modelu środowiska. Tworzy własną wewnętrzną reprezentację świata na podstawie danych wejściowych.
- Planowanie - MuZero wykorzystuje procesy podobne do wyszukiwania Monte Carlo Tree Search (MCTS), aby symulować różne scenariusze i przewidywać najbardziej korzystne działania. Wykorzystuje przy tym trzy modele:
  - Model przewidujący wartość bieżącego stanu.
  - Model przewidujący nagrodę za przejście w kolejny stan.
  - Model przewidujący następny stan na podstawie wykonanej akcji.
- Uczenie się z danych - System uczy się na podstawie danych historycznych oraz symulacji, aby ulepszać swoją strategię i lepiej przewidywać skutki akcji.
- Zastosowanie w różnych środowiskach - Może być stosowany w grach wideo, takich jak Atari, czy w bardziej złożonych środowiskach, takich jak Doom.



2 Informacje na temat eksperymentu

2.1 Literatura

Do zrozumienia problemu wykorzystano następujące pozycje:

1. `https://arxiv.org/pdf/1911.08265`
2. `https://www.youtube.com/watch?v=c8SLNEpFSrs`

2.2 Przydatne repozytoria

Przykładowe implementacje MuZero:

1. `https://github.com/johan-gras/MuZero`
2. `https://github.com/opendilab/LightZero`
3. `https://github.com/werner-duvaud/muzero-general`

Środowiska:

1. `https://github.com/Farama-Foundation/Arcade-Learning-Environment`
2. `https://paperswithcode.com/dataset/dqn-replay-dataset`
3. `https://github.com/clvrai/awesome-rl-envs`
4. `https://github.com/Farama-Foundation/ViZDoom`

2.3 Framework

Do powtórzenia eksperymentu będziemy korzystać z

1. pytorch
2. opencv
3. numpy
4. ray

2.4 Co będzie przedmiotem eksperymentu

Przeprowadzenie testów na

1. test na prostych środowiskach (cartpole, gridworld),
2. test na atari (atlantis, bowling, breakout, crazy climber, pacman, pong),
3. test na doom'ie.

2.5 Metryki i baseline

Wyuczenie agentów do poziomu przeciętnego człowieka, który grał parę godzin w grę.

Game	Human
Atlantis	29028.13
Bowling	160.73
Breakout	30.47
Crazy climber	35829.41
Pacman	6951.60
Pong	14.59

2.6 Sprzęt użyty w testach

Do testów wykorzystano 2 rodzaje komputerów:

1. Intel i7 13700k + 32GB ram + Nvidia RTX 4070 Ti,
2. Intel i5 4590 + 32GB ram.

2.7 Podział zadań

1. Marek - pisanie dokumentacji + trening/testy + integracja z grami
2. Paulina - implementacja modelu i pierwsze testy + wsparcie

3 Napotkane problemy

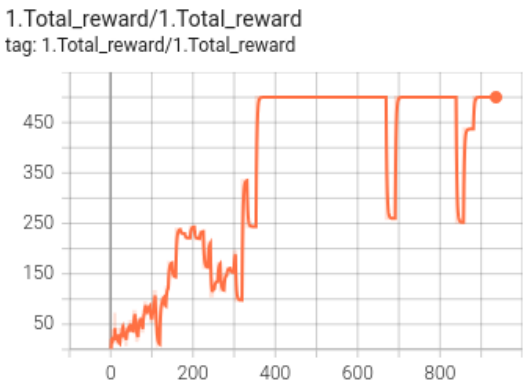
Podczas tworzenia sieci i przeprowadzania testów natrafiono na szereg problemów:

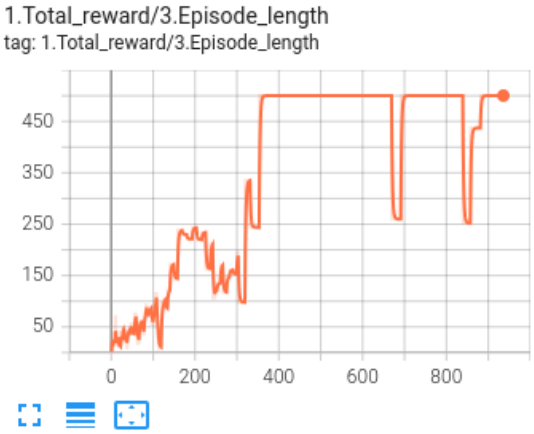
1. posiadanie nie zawsze spójnego środowiska,
2. Wyłączenie maszyn w losowym momencie,
3. Przygotowanie odpowiedniego zestawu pakietów pythonowych, aby środowisko mogło być automatyczne budowane.
4. Brak wymaganych bibliotek systemowych i brak odzewu w tej sprawie od zarządzających maszynami.
5. Częste przepełnienie ramu, brak możliwości puszczenia tego na odpowiednio mocnym sprzęcie - mocno ogranicza to model i wpływa na jego wyniki.
6. Nawet przy próbie pracy z tymi ograniczeniami, czasem model zapisywał się z błędem.

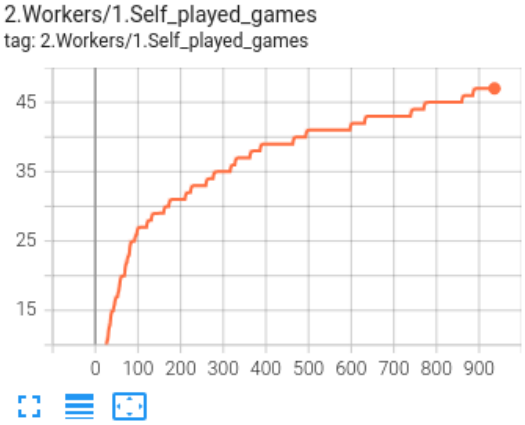
4 Wyniki

Zapis rozgrywek można obejrzeć w załączonym pliku zip z filmami.

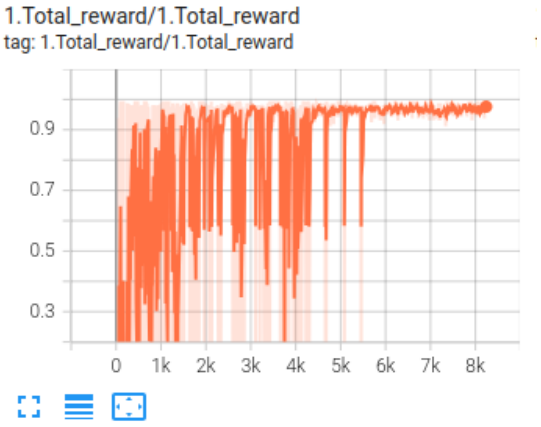
4.1 Cartpole

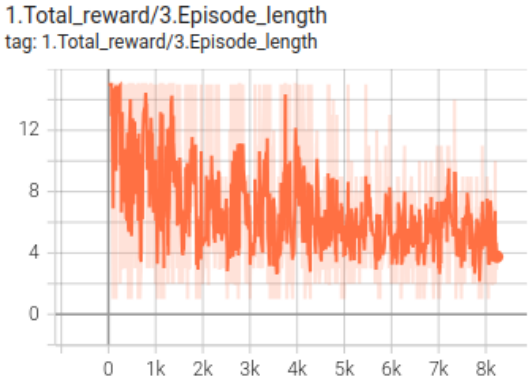

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
450
350
250
150
50
0
200
400
600
800

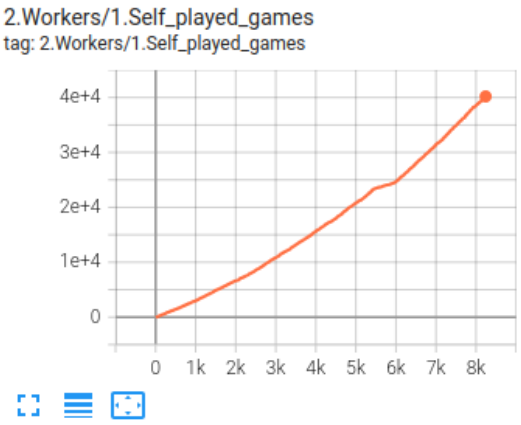

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
450
350
250
150
50
0
200
400
600
800


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
45
35
25
15
0
100
200
300
400
500
600
700
800
900

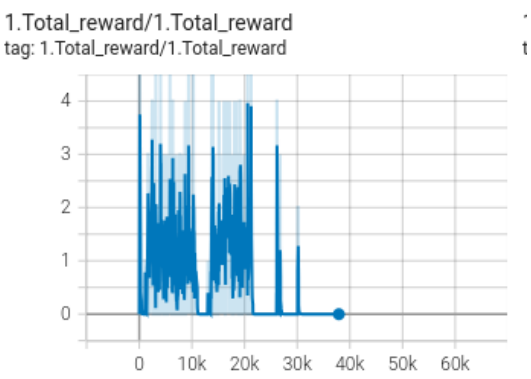
4.2 Gridworld

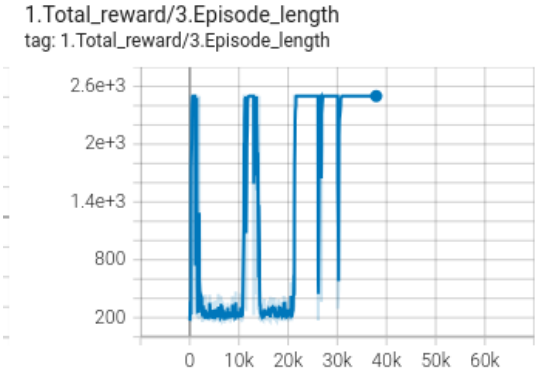


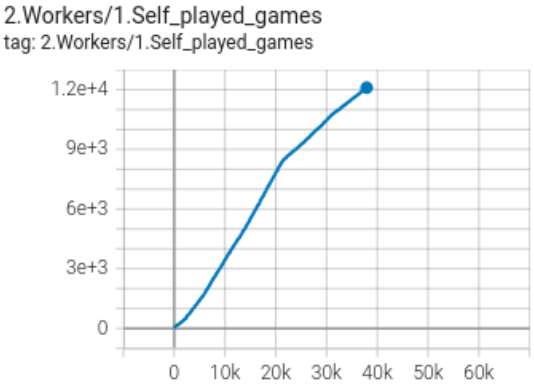




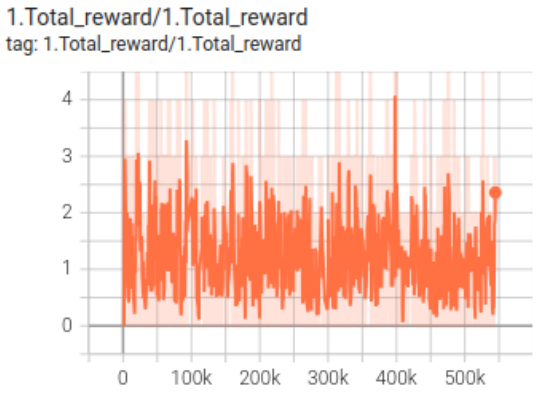
4.3 Breakout

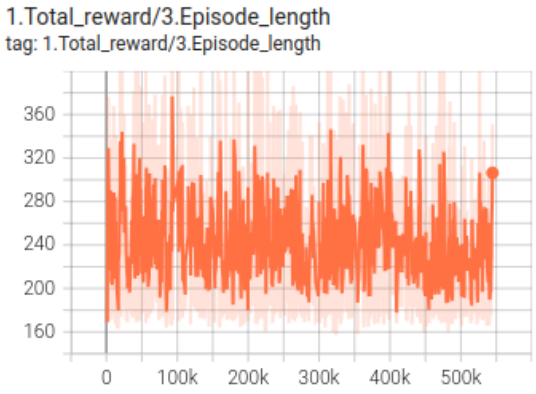

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
4
3
2
1
0
0
10k
20k
30k
40k
50k
60k

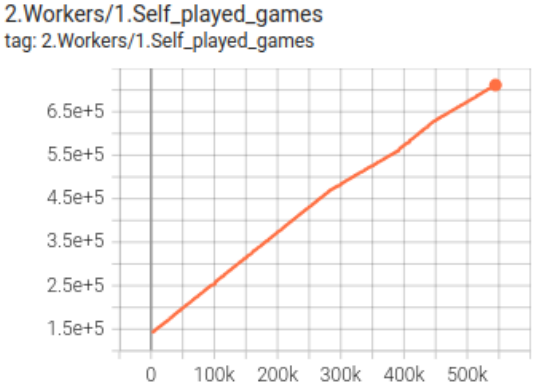

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
2.6e+3
2e+3
1.4e+3
800
200
0
10k
20k
30k
40k
50k
60k


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
1.2e+4
9e+3
6e+3
3e+3
0
0
10k
20k
30k
40k
50k
60k

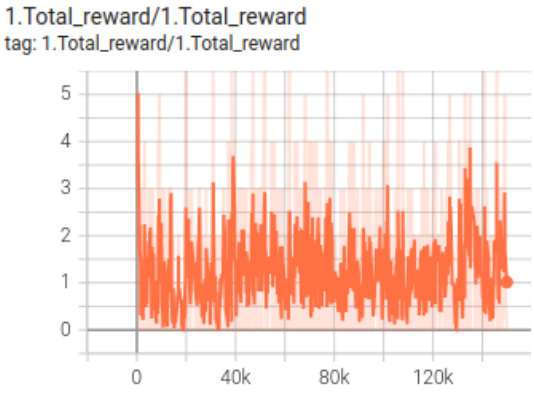
4.4 Breakout 5M

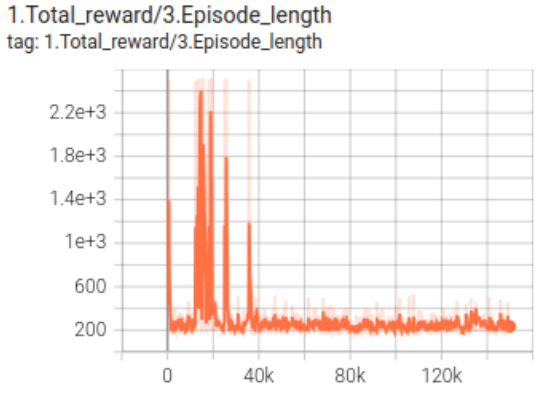


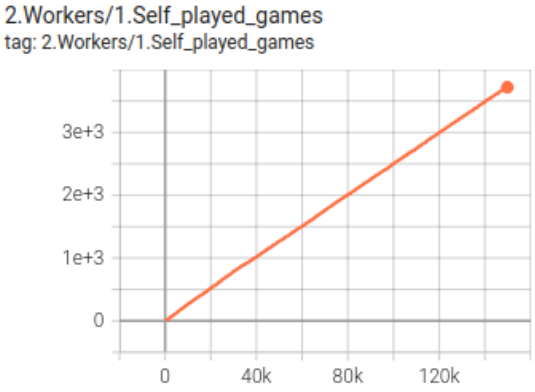




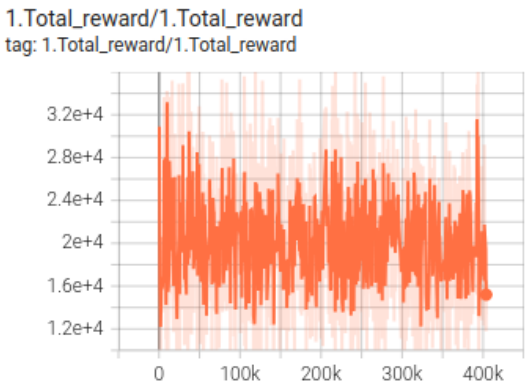
4.5 Breakout GPU



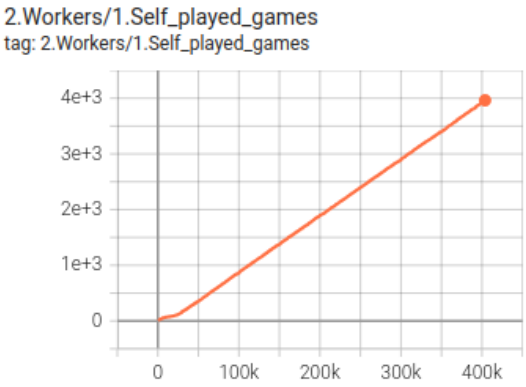




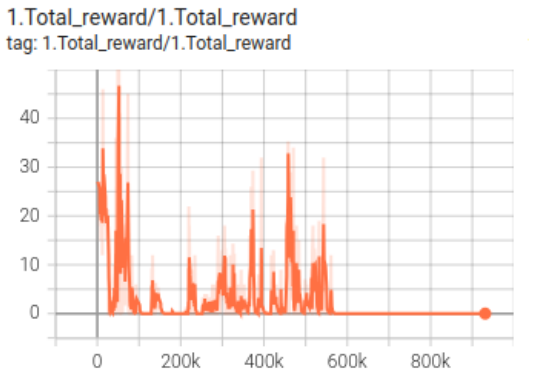
4.6 Atlantis

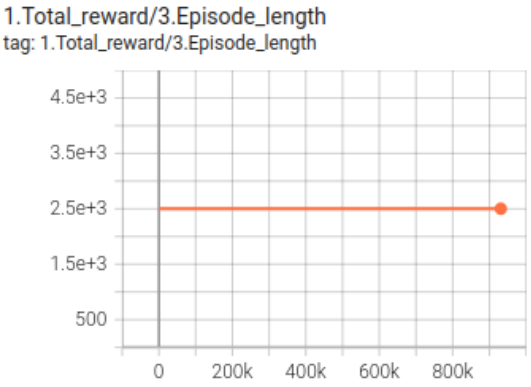

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
3.2e+4
2.8e+4
2.4e+4
2e+4
1.6e+4
1.2e+4
0
100k
200k
300k
400k

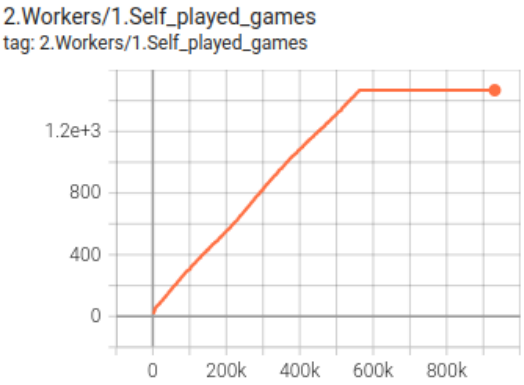

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
2.4e+3
2e+3
1.6e+3
1.2e+3
0
100k
200k
300k
400k


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
4e+3
3e+3
2e+3
1e+3
0
0
100k
200k
300k
400k

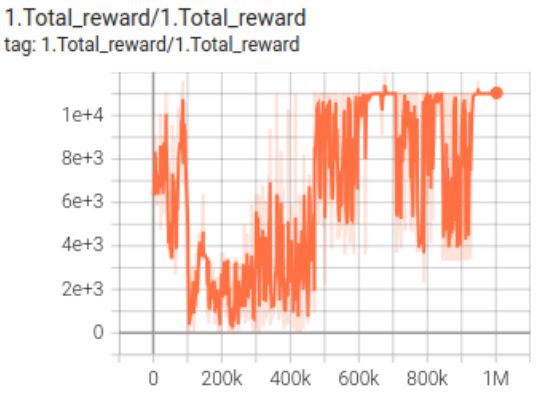
4.7 Bowling

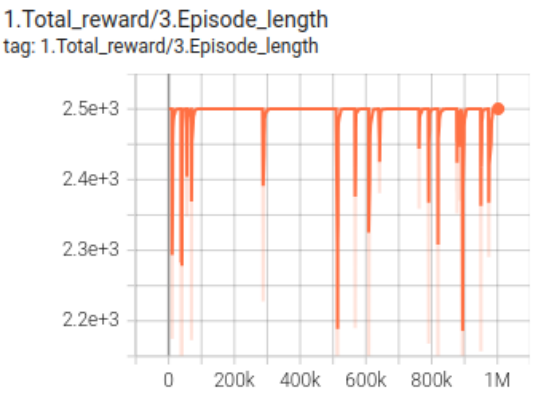

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
40
30
20
10
0
0
200k
400k
600k
800k

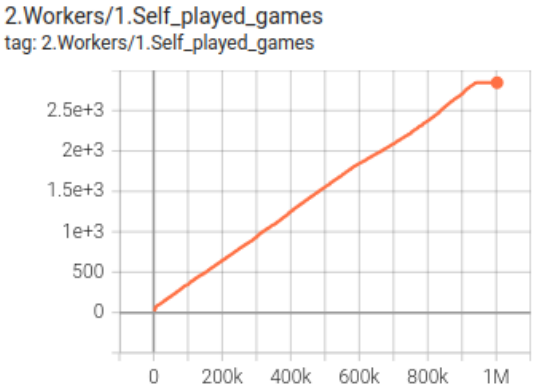

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
4.5e+3
3.5e+3
2.5e+3
1.5e+3
500
0
200k
400k
600k
800k


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
1.2e+3
800
400
0
0
200k
400k
600k
800k

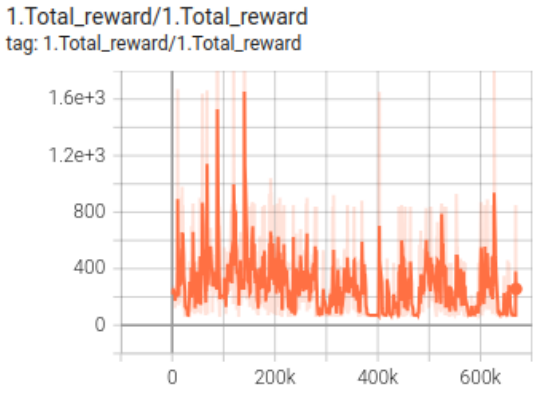
4.8 Crazy climber

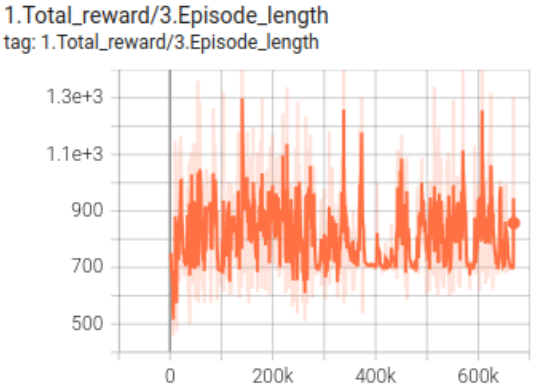


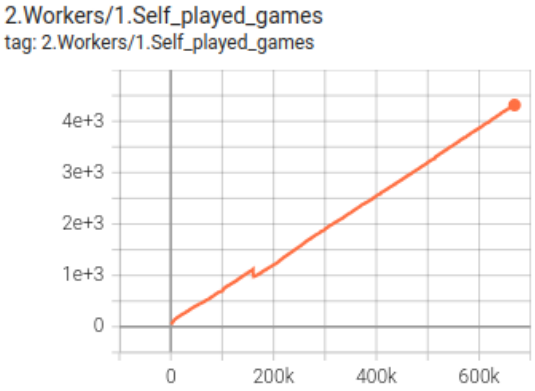




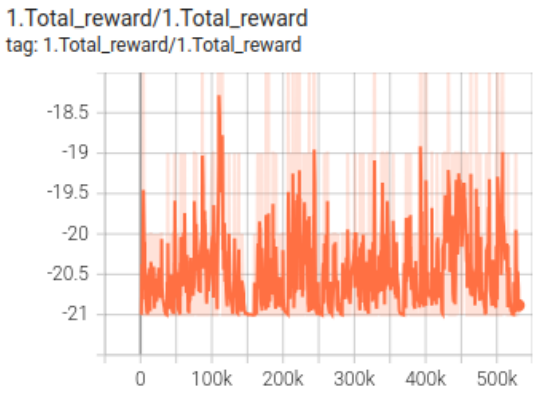
4.9 Pacman

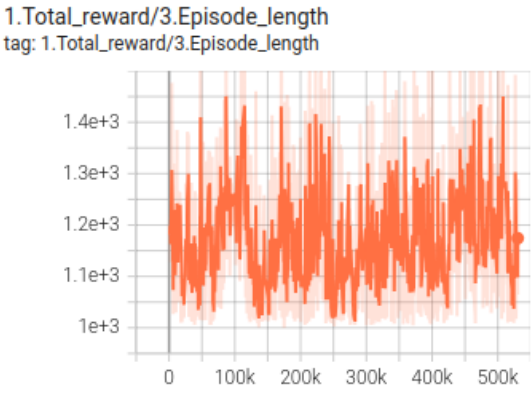

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
1.6e+3
1.2e+3
800
400
0
0
200k
400k
600k

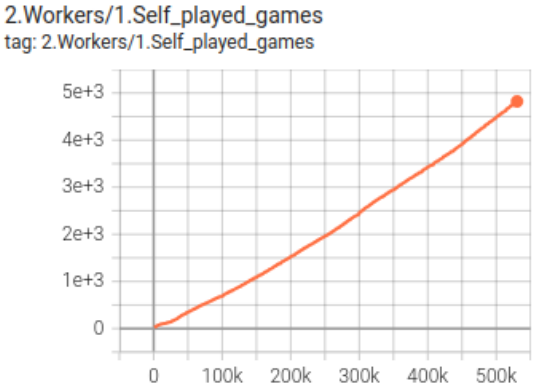

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
1.3e+3
1.1e+3
900
700
500
0
200k
400k
600k


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
4e+3
3e+3
2e+3
1e+3
0
0
200k
400k
600k

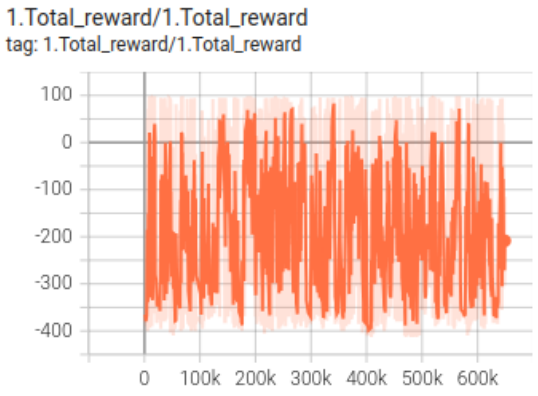
4.10 Pong

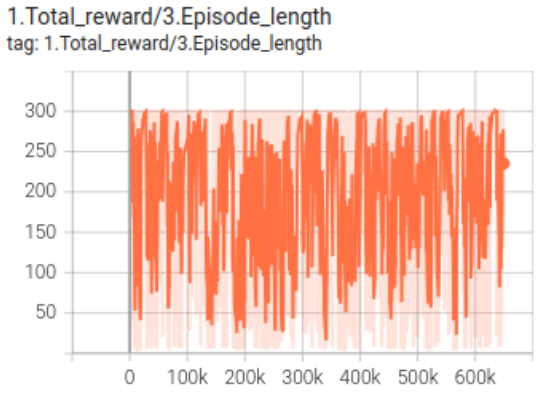

1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward
-18.5
-19
-19.5
-20
-20.5
-21
0
100k
200k
300k
400k
500k

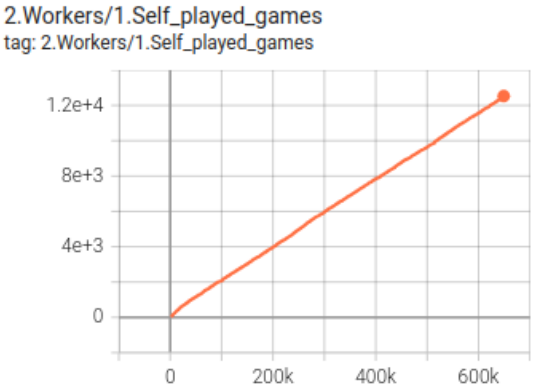

1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length
1.4e+3
1.3e+3
1.2e+3
1.1e+3
1e+3
0
100k
200k
300k
400k
500k


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games
5e+3
4e+3
3e+3
2e+3
1e+3
0
0
100k
200k
300k
400k
500k

4.11 Doom







5 Podsumowanie

Mimo że sam pomysł muzero ma bardzo duży potencjał, badaczom nie udało się go w pełni odtworzyć. Większość problemów napotkanych w odtwarzaniu można przypisać niedoborom sprzętowym towarzyszącym podczas prób uczenia. Widać jednak, że metoda ta ma zastosowanie, chociażby w cartpole, czy gridworld, które aż tak dużych wymagań co do danych nie miały. Najbardziej obiecującą okazały się gry crazy climber i atlantis, gdzie agenci wykazywali się pewną inteligencją w ruchach.