

Czy na MuZero można odpalić Doom’a?

Paulina Brzęcka

Marek Borzyszkowski

23 stycznia 2025

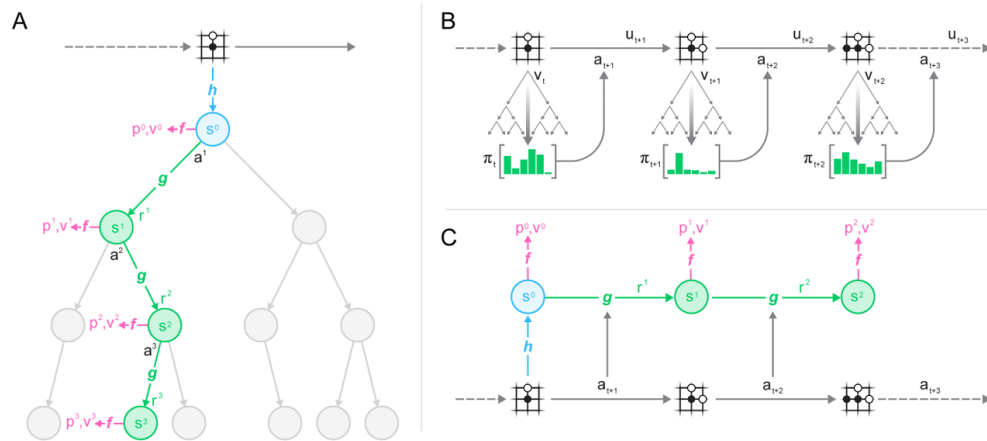
Spis treści

| | | |
|----------|--|-----------|
| 1 | Jak działa MuZero | 2 |
| 2 | Informacje na temat eksperymentu | 2 |
| 2.1 | Literatura | 2 |
| 2.2 | Przydatne repozytoria | 3 |
| 2.3 | Framework | 3 |
| 2.4 | Co będzie przedmiotem eksperymentu | 3 |
| 2.5 | Metryki i baseline | 3 |
| 2.6 | Sprzęt użyty w testach | 4 |
| 3 | Napotkane problemy | 4 |
| 4 | Wyniki | 4 |
| 4.1 | Cartpole | 5 |
| 4.2 | Gridworld | 6 |
| 4.3 | Breakout | 7 |
| 4.4 | Breakout 5M | 8 |
| 4.5 | Breakout GPU | 9 |
| 4.6 | Atlantis | 10 |
| 4.7 | Bowling | 11 |
| 4.8 | Crazy climber | 12 |
| 4.9 | Pacman | 13 |
| 4.10 | Pong | 14 |
| 4.11 | Doom | 15 |
| 5 | Podsumowanie | 15 |

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/opensilab/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,
2. test na atari,
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.

3 Napotkane problemy

Podczas tworzenia sieci i przeprowadzania testów natrafiono na szereg problemów:

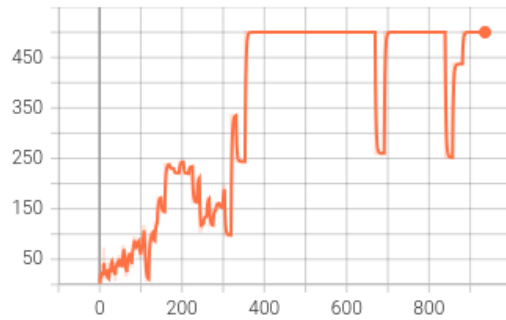
1. posiadanie nie zawsze spójnego środowiska,
2. Wyłączenie maszyn w losowym momencie,
3. Przygotowanie odpowiedniego zestawu pakietów pythonowych, aby środowisko mogło być automatycznie budowane.
4. Brak wymaganych bibliotek systemowych i brak odzewu w tej sprawie od zarządzających maszynami.
5. Częste 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

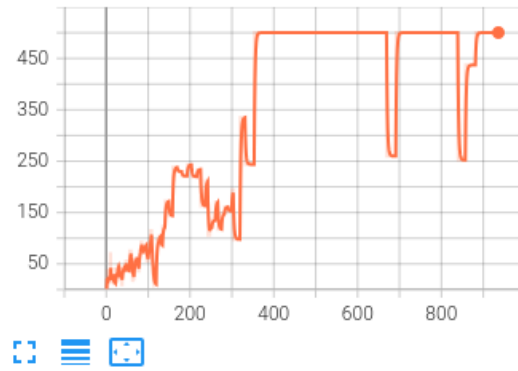
Zapis rozgrywek można obejrzeć w załączonym pliku zip z filmami.

4.1 Cartpole

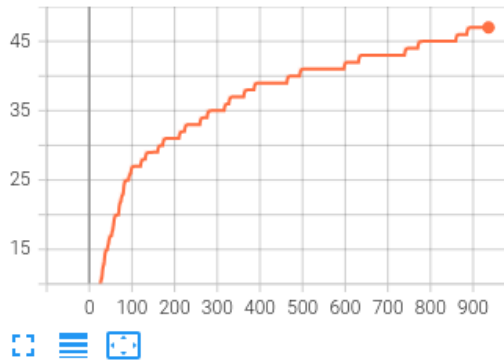
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

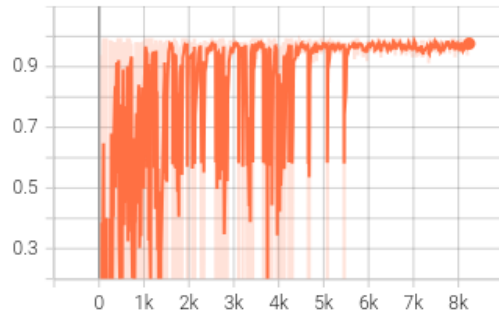


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

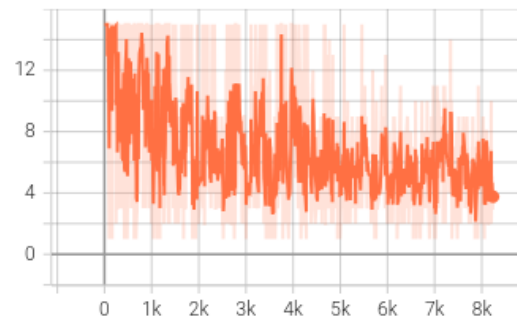


4.2 Gridworld

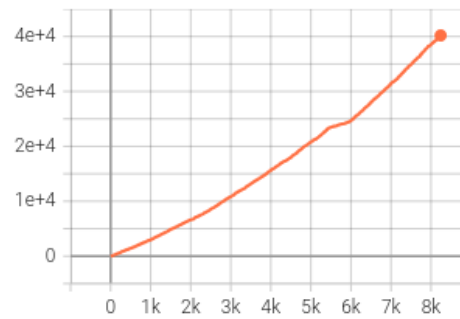
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

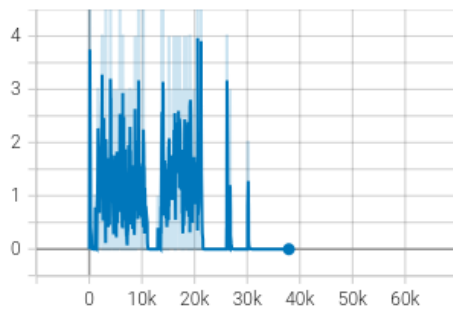


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

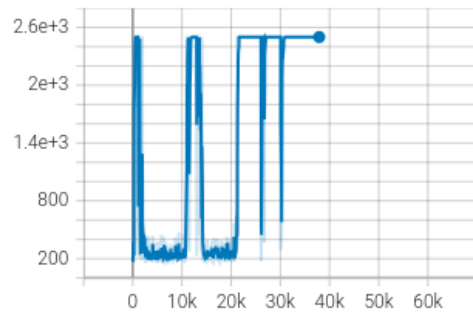


4.3 Breakout

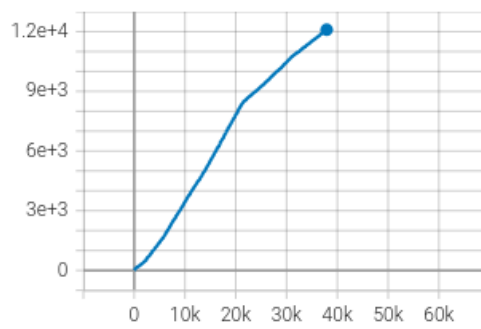
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

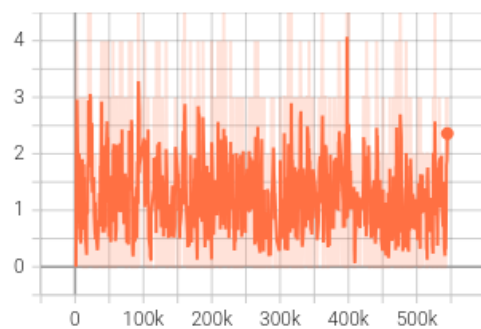


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

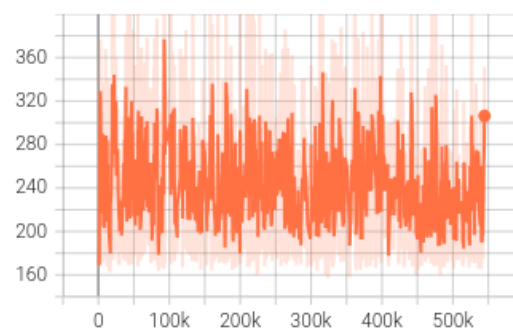


4.4 Breakout 5M

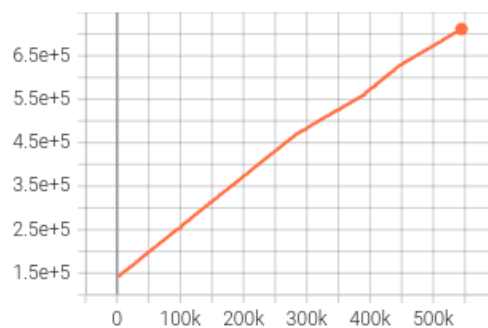
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

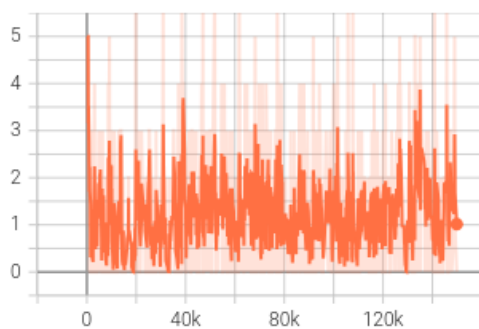


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

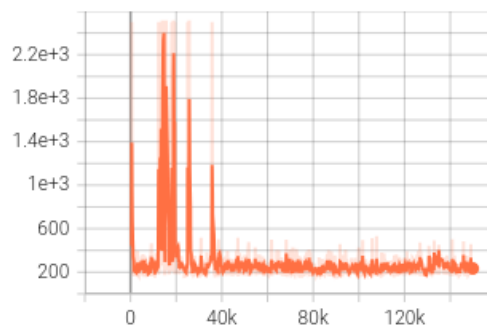


4.5 Breakout GPU

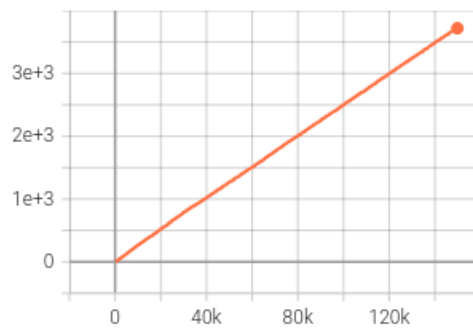
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

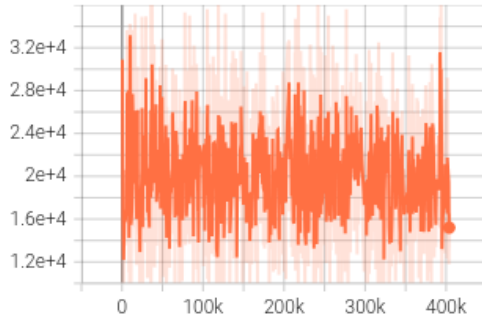


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

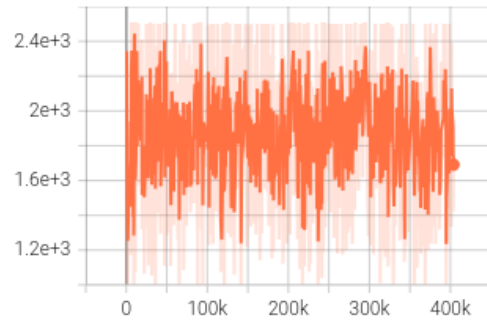


4.6 Atlantis

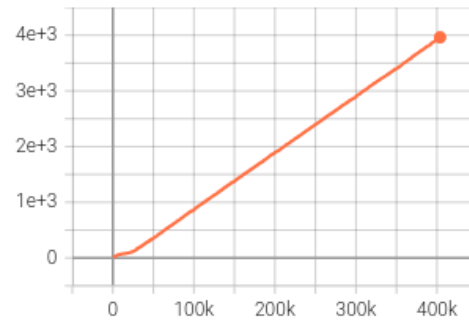
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

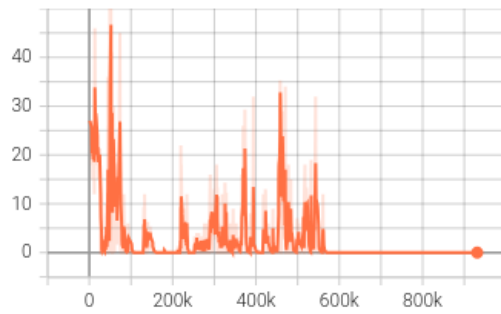


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

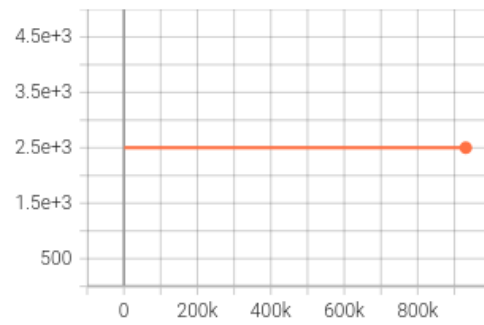


4.7 Bowling

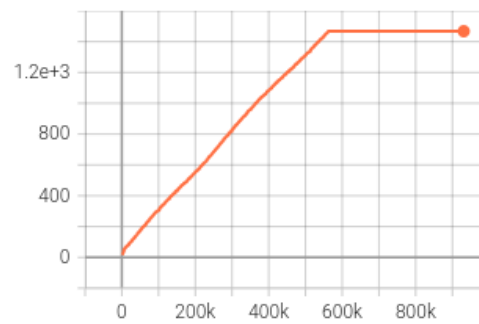
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

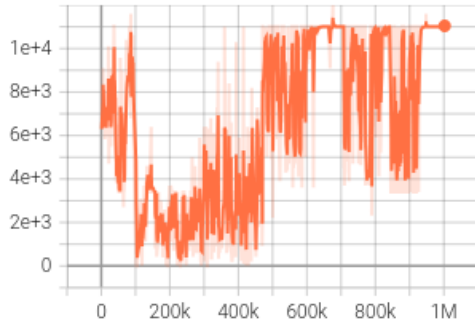


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

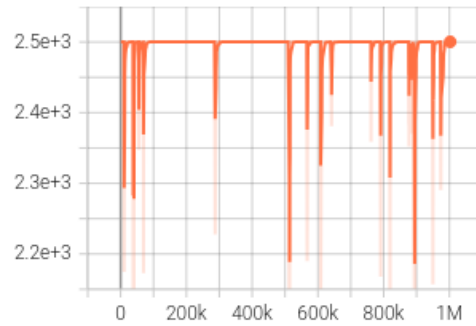


4.8 Crazy climber

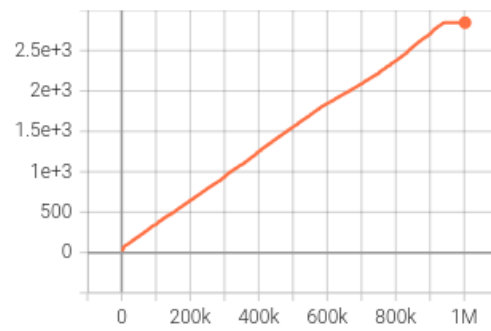
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

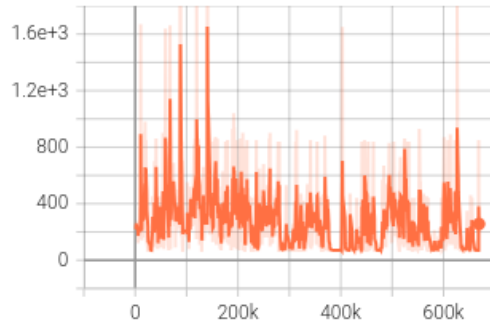


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

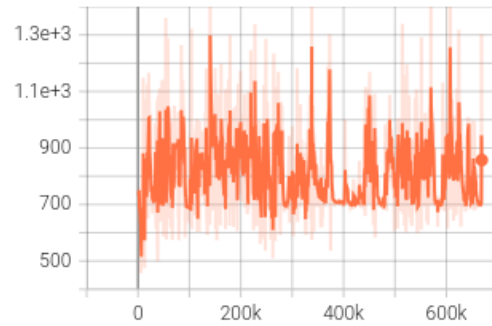


4.9 Pacman

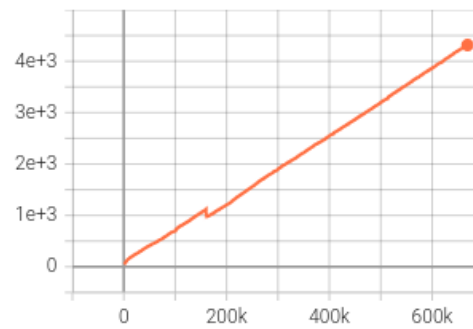
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

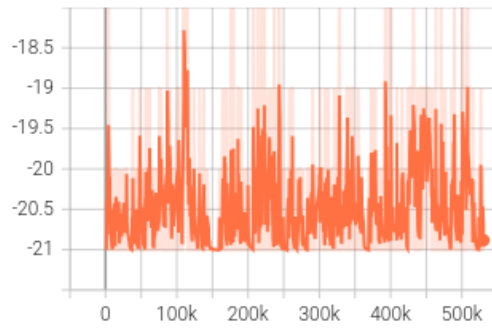


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

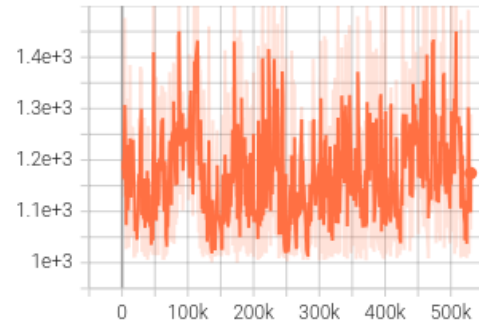


4.10 Pong

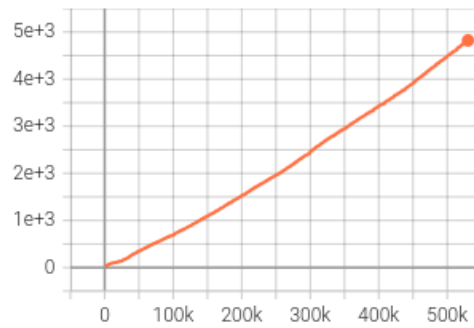
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length

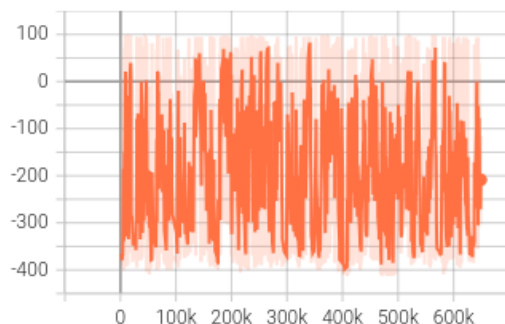


2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games

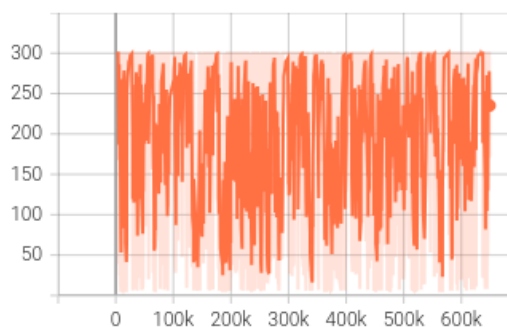


4.11 Doom

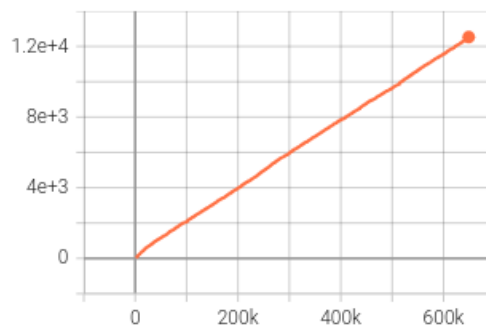
1.Total_reward/1.Total_reward
tag: 1.Total_reward/1.Total_reward



1.Total_reward/3.Episode_length
tag: 1.Total_reward/3.Episode_length



2.Workers/1.Self_played_games
tag: 2.Workers/1.Self_played_games



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.