Asynchronous Deep Learning Methods for *Super Mario Bros.*

Matthew Jones and Campbell Sinclair

Super Mario Bros. - Background

- 2D platformer for the NES
- State is an RGB frame of the game, actions are NES controller buttons
- Reward is based on distance traveled to the right, score, time, and whether the level is beaten

Method

- Resize images to 84x84 then grayscale, stack, and normalize the images
- We implemented three asynchronous variants of algorithms we learned in class [Mnih et al, 2016]:
  - 1-step Q-learning
  - N-step Q-learning
  - Asynchronous Advantage Actor-critic (A3C)

Asynchronous Algorithms

- Thread-specific environment and parameters
- Act using thread-specific parameters
- Periodically copy global parameters to thread-specific parameters
- Each learner accumulates gradients using the same target as their respective non-asynchronous algorithm

Experiments and Results

- We trained our three implementations on World 1-1 for 6 million steps
- Compared to a DDQN implementation and random agent as baselines
- Evaluated the best agent produced by each method

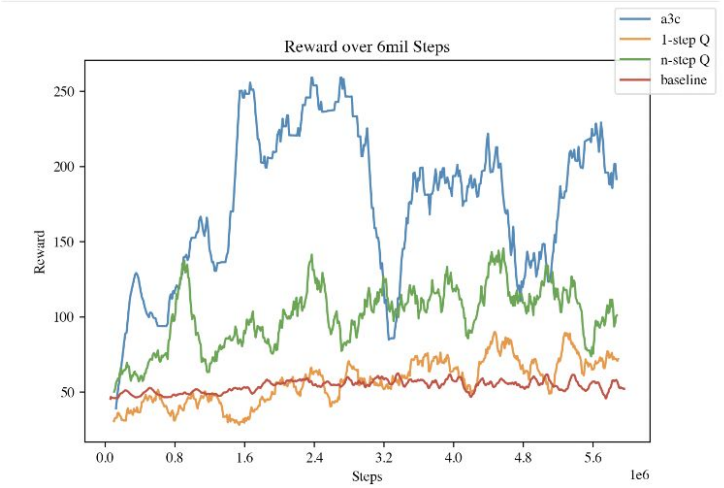


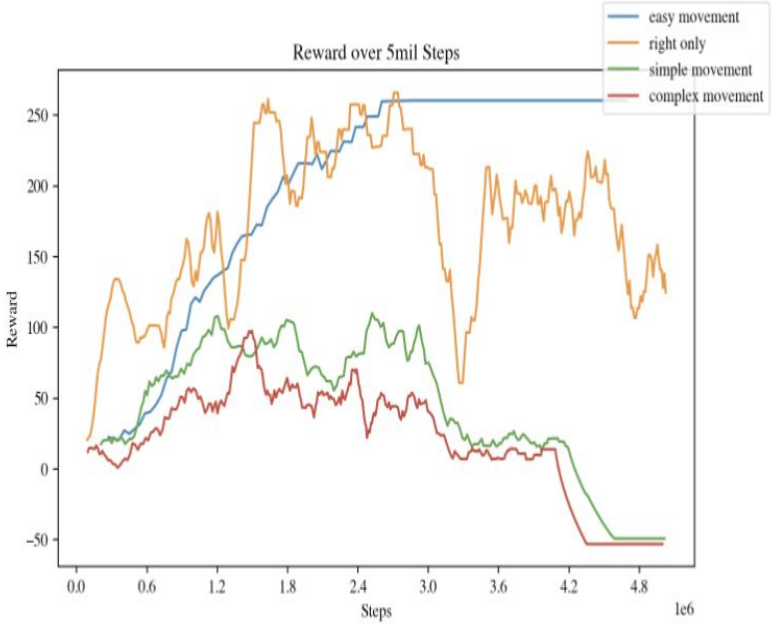
Figure 2: Training performance on level 1 using various algorithms

Algorithm	Average Score	First level wins
A3C	271.33	100
1-step Q-learning	47.65	0
N-step Q-learning	121.40	16
Pytorch Tutorial DDQN	59.27	0
Random Agent	44.85	0

Table 1: Average score over 100 episodes of the best model from training.

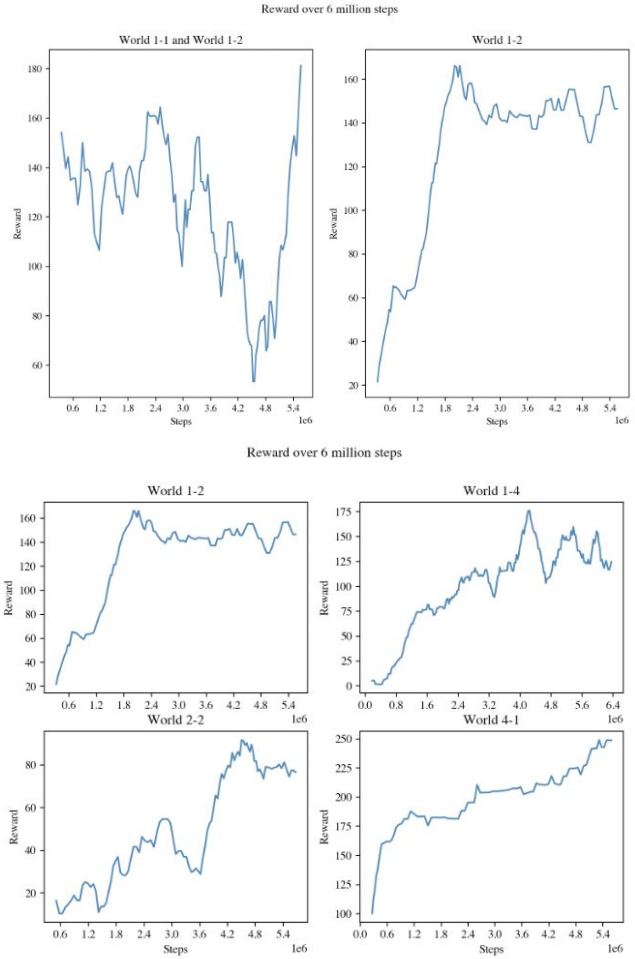
Experiments and Results

- Used our A3C implementation on World 1-1
- Varied the set of actions available to the agent

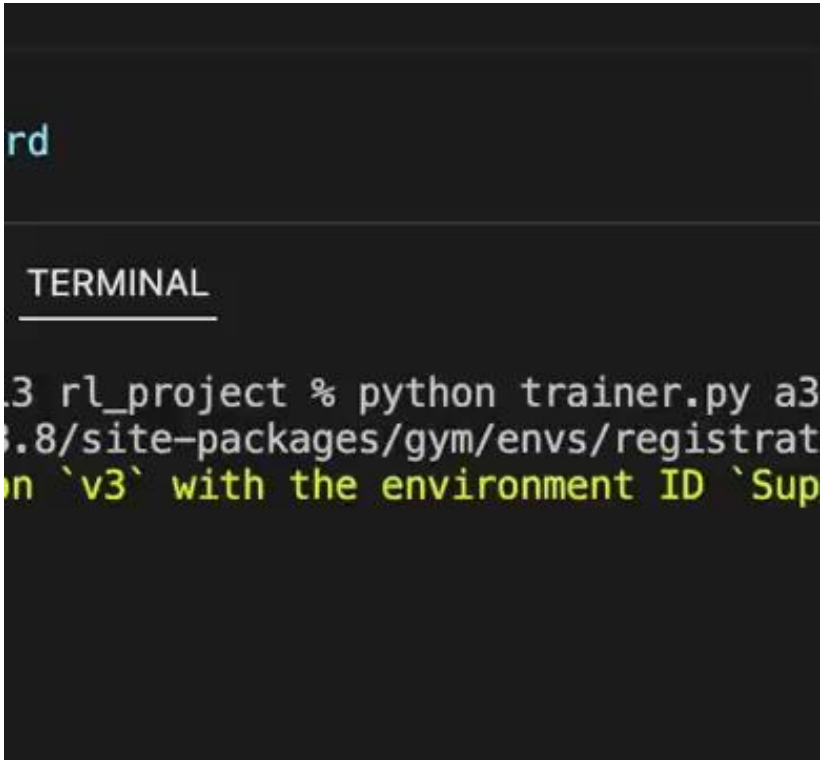


Experiments and Results

- Tested our A3C implementation on different levels of the game
- It is difficult for our agent to learn more than 1 level with the same set of parameters



Demo

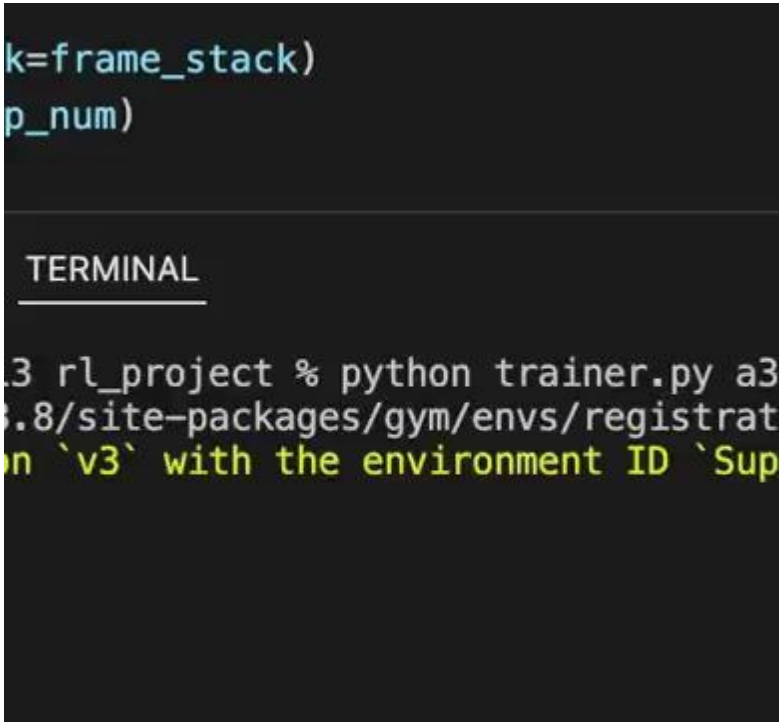

rd
TERMINAL
3 rl_project % python trainer.py a3
.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup

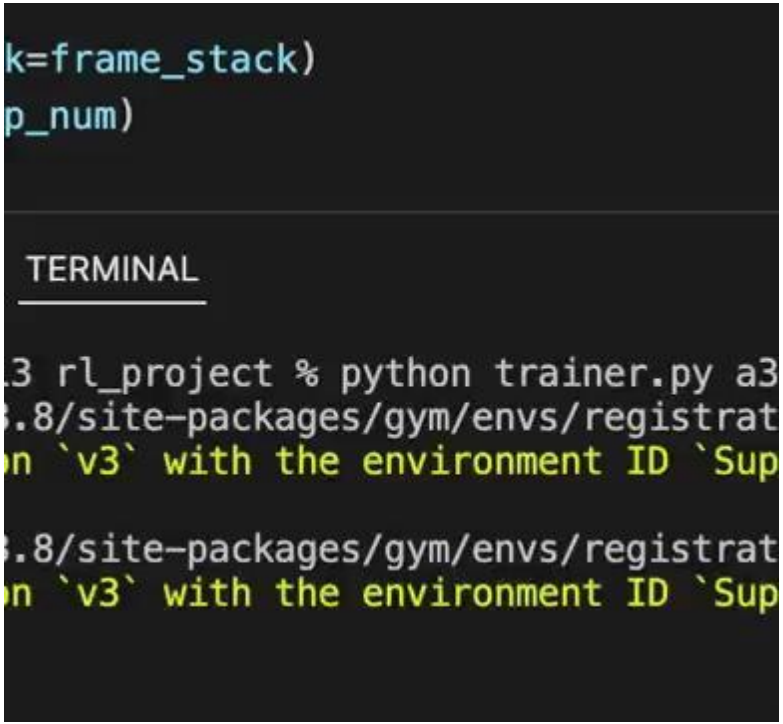

MARIO
000000
×00
WORLD
1-1
TIME
397

Demo

```
k=frame_stack)
p_num)

TERMINAL

3 rl_project % python trainer.py a3
.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup

.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup
```

Demo


k=frame_stack)
p_num)
TERMINAL
3 rl_project % python trainer.py a3
.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup


k=frame_stack)
p_num)
TERMINAL
3 rl_project % python trainer.py a3
.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup
.8/site-packages/gym/envs/registrat
n `v3` with the environment ID `Sup

References

Recht, Benjamin / Re, Christopher / Wright, Stephen / Niu, Feng(2011): *Hogwild!: A lock-free approach to parallelizing stochastic gradient descent.*

Mnih, Volodymyr u.a.(2015): *Human-level control through deep reinforcement learning*, 7540: 529-533.

Mnih, Volodymyr / Badia, Adrià Puigdomènech / Mirza, Mehdi / Graves, Alex / Lillicrap, Timothy P. / Harley, Tim / Silver, David / Kavukcuoglu, Koray(2016): *Asynchronous Methods for Deep Reinforcement Learning*.

Arras, Leila / Arjona-Medina, Jose A. / Widrich, Michael / Montavon, Grégoire / Gillhofer, Michael / Müller, Klaus-Robert / Hochreiter, Sepp / Samek, Wojciech(2019): *Explaining and Interpreting LSTMs*.

Feng, Yuansong / Subramanian, Suraj / Wang, Howard / Guo, Steven (2022): *Training a Mario-Playing RL Agent*
`https://pytorch.org/tutorials/intermediate/mario_rl_tutorial.html`.

Kauten, Christian (2018): *Super Mario Bros for OpenAI Gym*
https://github.com/Kautenja/gym-super-mario-bros.

Kingma, Diederik P. / Ba, Jimmy (2014): *Adam: A Method for Stochastic Optimization*
.

Kostrikov, Ilya (2018): *PyTorch Implementations of Asynchronous Advantage Actor Critic*
`https://github.com/ikostrikov/pytorch-a3c`.

mariowiki.com (2022): *Super Mario Bros.*
`https://www.mariowiki.com/Super_Mario_Bros.`

Sadeq, Amine / Sakhi, Otmane (2019): *Exploring Deep Reinforcement Learning with Super Mario Bros*
`https://github.com/sadeqa/Super-Mario-Bros-RL`.

Sutton, Richard S / Barto, Andrew G (2018): *Reinforcement learning: An introduction.* , MIT press.

Zhang, Shangtong (2018): *DeepRL*
`https://github.com/ShangtongZhang/DeepRL/releases/tag/v0.1`.