Change in Behavior and Transfer with Change in Properties of the Environment

Aditya Anantwar Abhinandan De

Instructor: Prof. Aritra Hazra

IIT Kharagpur

Term Project

November 11, 2022

Presentation Overview

1 Introduction

2 Methods
- Algorithms
- General Methodology

3 Observations
- DQN
- PPO

4 Conclusion

Introduction

- Aim of project is to analyze change in behaviour of RL based systems with change in properties.
- We utilize OpenAI's CartPole environment [1] for analysis.

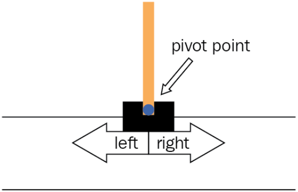

Figure: The CartPole problem.

Algorithms

Q-Learning

- Model free algorithm [3]
- State action pair backups

$$Q_\theta(S_t, A_t) \leftarrow Q_\theta(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q_\theta(S_{t+1}, a, \theta) - Q_\theta(S_t, A_t)]$$

DQN

- A DQN approximated the state value function in a Q-Learning framework using neural networks. [2]
- It uses **Experience Replay** to prevent overfitting and a **Target Network** to improve training efficiency.

Algorithms

PPO

- Proximal Policy Optimization is the policy gradient method where the policy is updated explicitly.
- The objective function is given by:

$$J(\theta) = E[\min(r(\theta)A_{\theta_{old}}(s,a), clip(r(\theta), 1-\epsilon, 1+\epsilon)A_{\theta_{old}}(s,a))]$$

- The clip function truncates the policy ratio between the range [1 - ϵ, 1 + ϵ]

DQN Observations

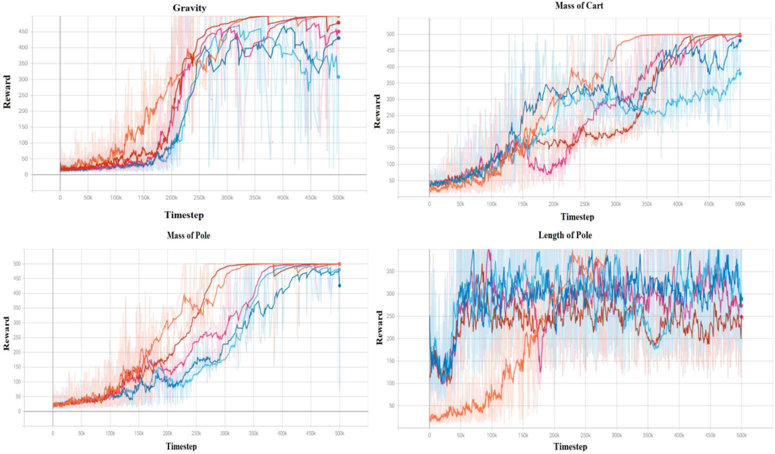

Figure: Reward vs Time steps for runs of DQN where the attributes of the environment were varied. In most cases, the network reaches the max reward by the end..

Example

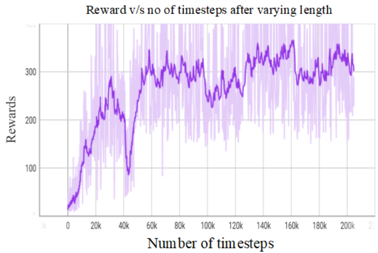

Figure: Transfer learning with cold start. Notice the sudden decrease around 40K and around 160K steps. There are subtle changes around 80K and 120K steps as well. The decrease shows us the impact of the configuration change on our policy.

[2]

More examples

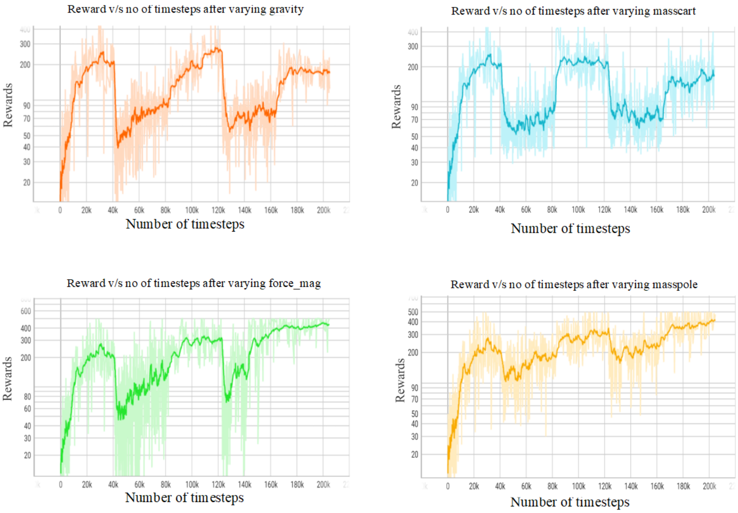

Figure: Configurations trained using PPO in a transfer learning setup.

Transfer Learning: Results

We now try and compare between vanilla policy gradient and with our changing environment under two scenarios: Zero shot and cold start

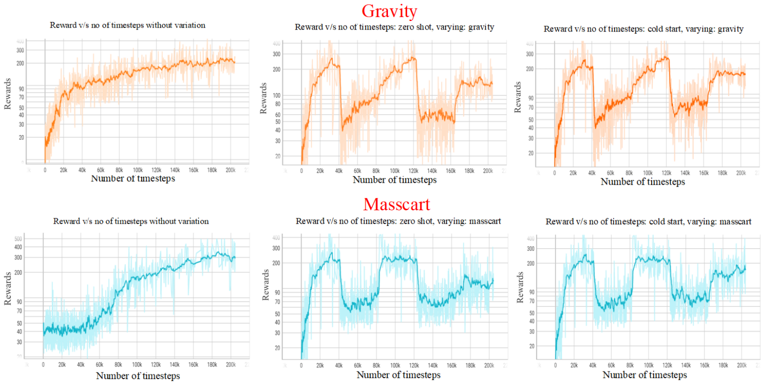

Figure: The difference between the second and the third columns arises from step=120K, this is when we stop back prop and perform a zero shot transfer.

Improvement!

The cold start performs better than vanilla policy gradient!

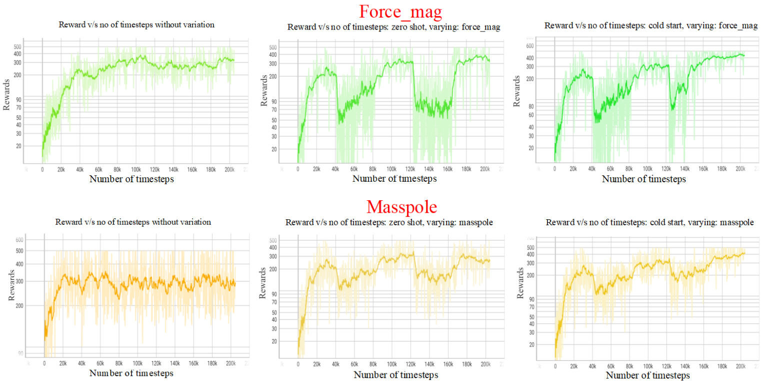

Figure: A tough initial phase might be beneficial!.

Conclusion

- We extensively evaluate different RL algorithms right from tabular methods based on Bellman backups to value function approximation and advanced Policy Gradient methods
- We find DQN to have a smoothening factor. This comes in via experience replay.
- We find transfer learning to be useful to generalize to new environments.
- In general solving for tougher environments in the beginning may turn out to be useful.

References

[1] Greg Brockman et al. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[2] Shengyi Huang et al. "CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms". In: *Journal of Machine Learning Research* 23.274 (2022), pp. 1–18. URL: http://jmlr.org/papers/v23/21-1342.html.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

The End

Thank You!