Parallel Double Cart Pole

UCLA 239AS Project Poster S2021

Sunay Bhat, Samuel Gessow, Vahe Gyuloglyan, Yi-Chun Hung

Background

- Novel extension of the traditional cart pole environment to expand on dynamics [1][3]
  - **Two Carts, Three Poles, Three Joints, 1 Pendulum**
- Dynamics of the model vastly increase the non-linearity and thus, difficulty of the controls problem[3]
- Possibility for multi-agent extensions

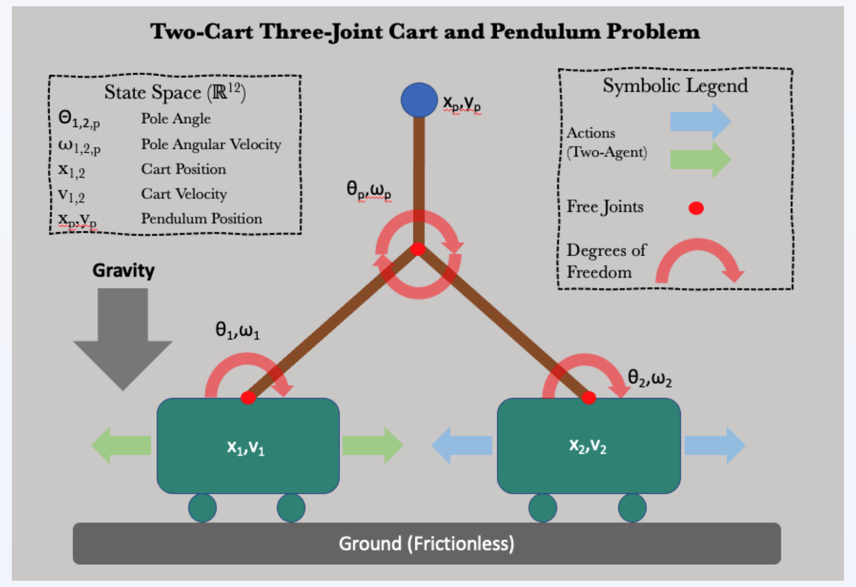


Figure 1. Double Cart pole environment with 12 environment state variables and 3x3 actions

Environment Implementation

- OpenAI Gym environment was implemented using Pymunk for the the physics simulation:
  - **Force Magnitude: 5, Time Step: 10ms (100 Hz)**
- Reward Structure:
  - Time Step Itself **(1/100)**
  - **x2** if time >10s, **x10** if time>100s
- Done Constraints[3]:
  - Upper pendulum must be **within pi/8 radians from vertical**
  - **Carts must be within 2.5 meters** from starting position
  - y-coord of the pendulum pole needs to be **above .15 meters**
- Goal: Stay within constraint bounds for 200 seconds for the 'vast majority of starting angles between -12 and 12 degrees' of center
  - We define this as a **'mean result >100 seconds' in the -12 to 12 degree range**

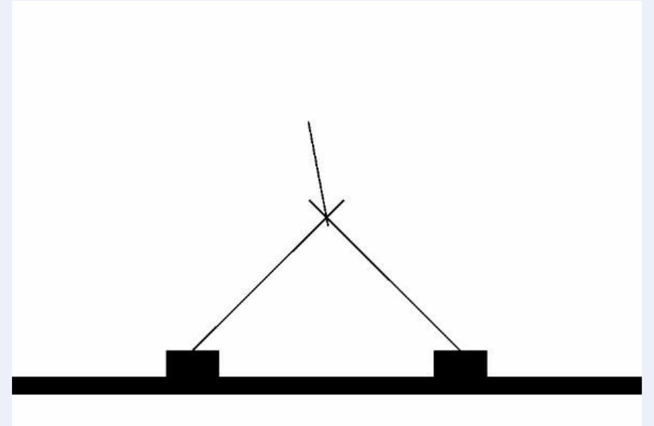


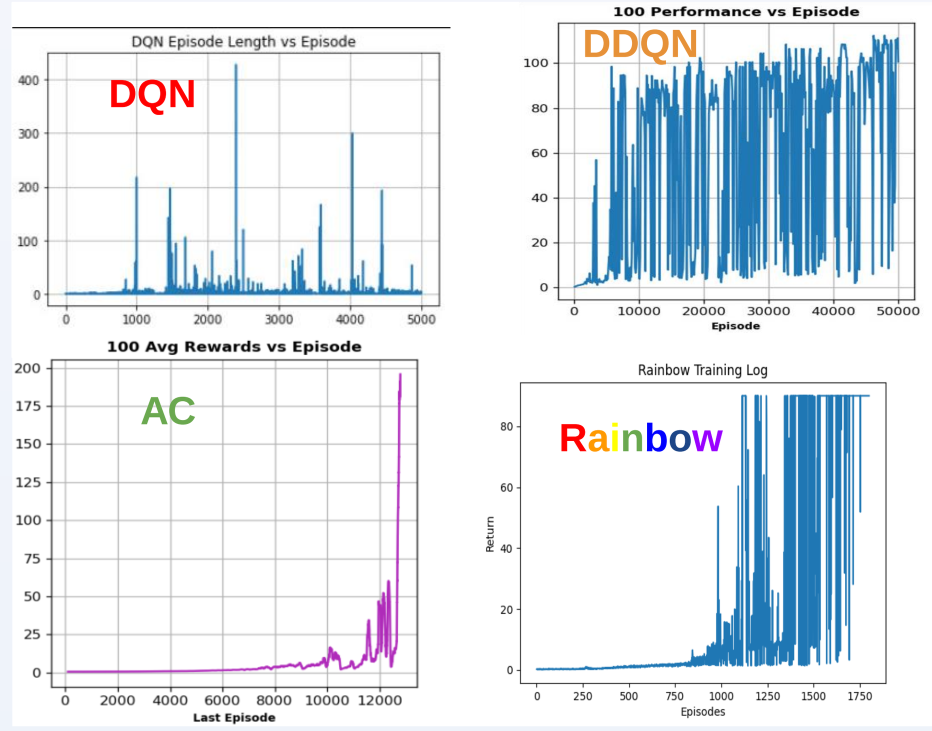
Figure 2. Environment render

Agent Algorithmic and Model Implementations

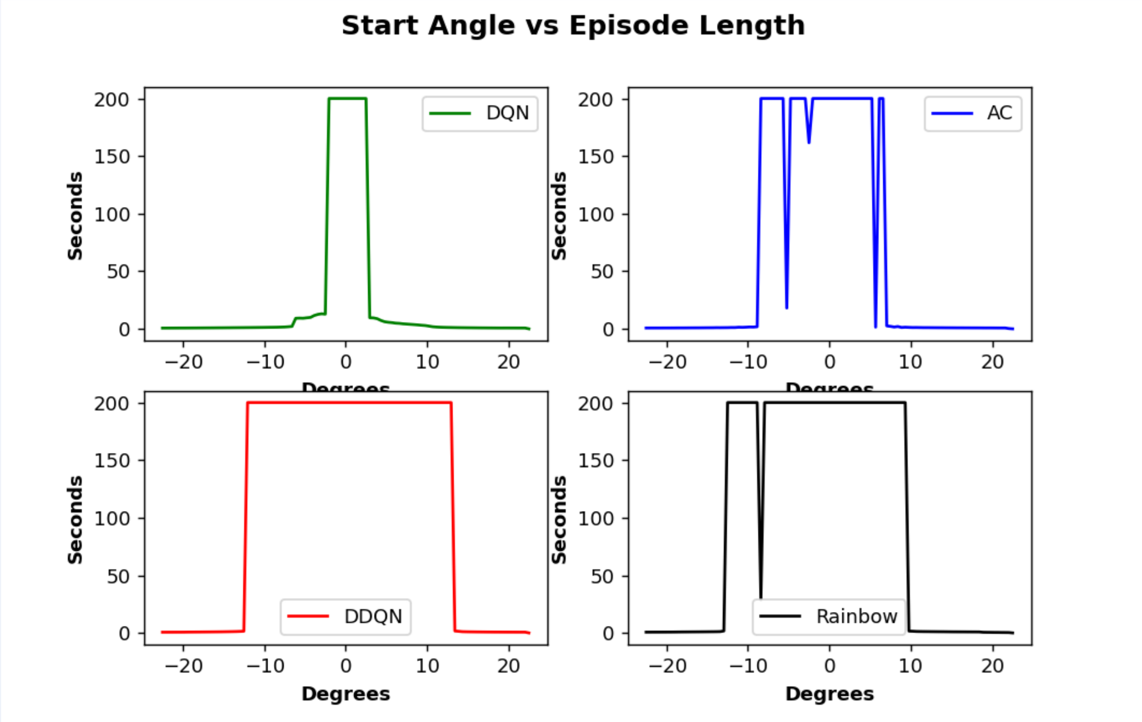
DQN[4]	DDQN[5]	Actor Critic[2]	Rainbow[7]
DQN: 12x144x9* *[**|S|** x **hidden1** x **|A|**] ***hidden layers**: sequential with fully connected linear, batch norm, rectifier	**DQN 1**: 12x64x9* **DQN 2**: 12x64x9* *[**|S|** x **hidden1** x **|A|**] ***hidden layers**: linear fully connected	**Actor**: 12x128x256x9* **Critic**: 12x128x256x1* *[**|S|** x **hidden1** x (**|A| or V**)] ***hidden layers**: linear fully connected	**DQN 1**: 12x512x512x9* **DQN 2**: 12x512x512x9* *[**|S|** x **hidden1** x **|A|**] ***hidden layers**: linear fully connected
DQN Agent uses a replay memory of 50k with a batch size of 512[7] to convert state input into action output	**Two DQN** networks are used alternatively to select and evaluate an action, same parameters as the DQN agent.	**Actor**: Returns action values ***using a categorical distribution*** **Critic**: Returns state value	Rainbow agent is a combination of DDQN, Prioritized Replay, Actor Critic, Noisy, and Distributional DQN[7]

Results

Training



Testing

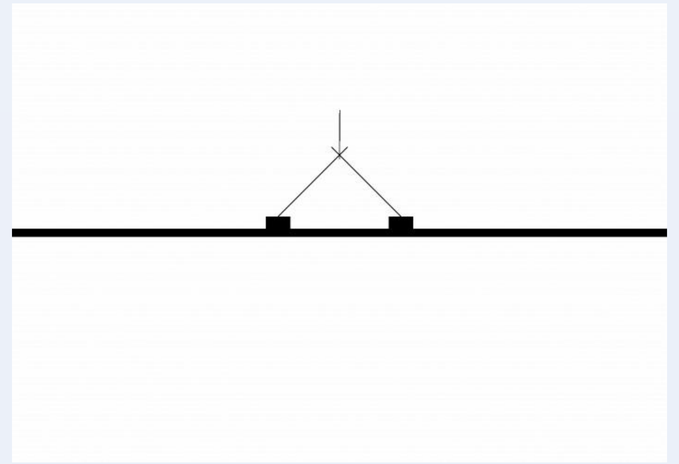


Mean Episode Length -12:12 Degrees

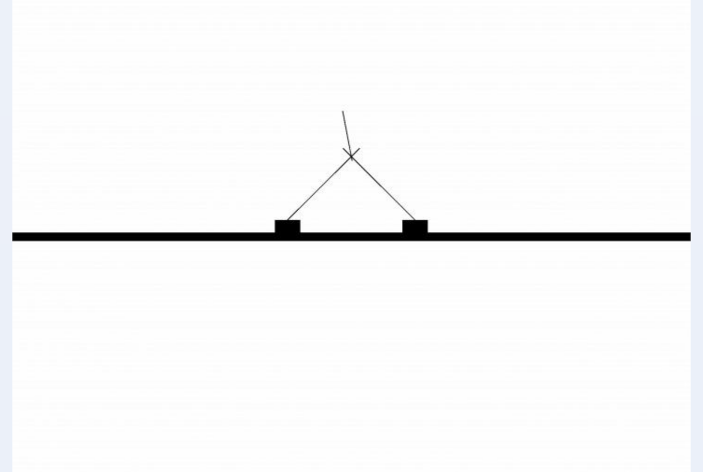
DQN[4]	DDQN[5]	Actor Critic[2]	Rainbow[7]
46.34	200	123.24	177.61*

* Note the Rainbow performance window is shifted with a left bias, the width of max episodes is similar to rainbow

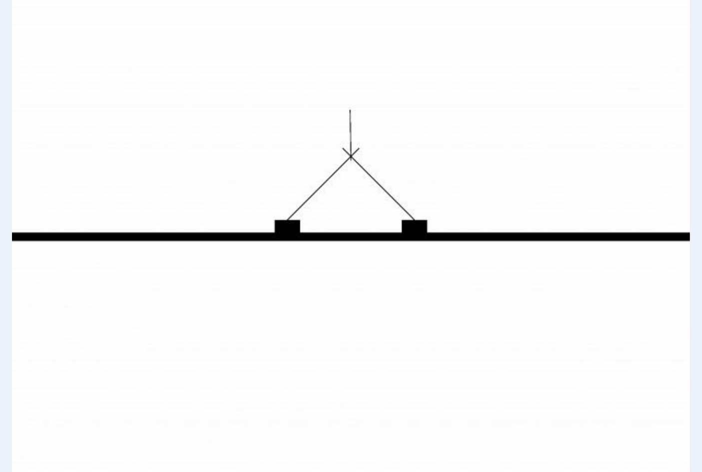
Actor-Critic Video



DDQN Video



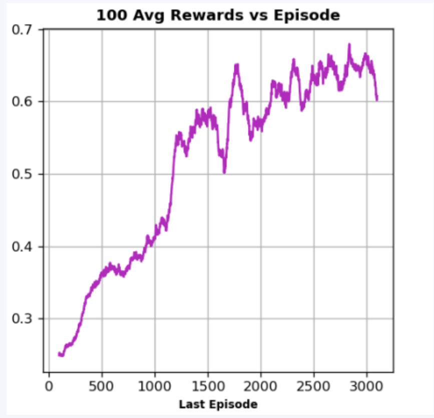
Rainbow Video

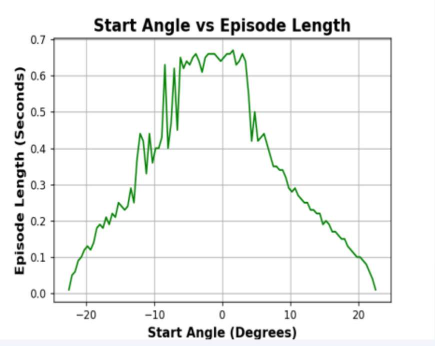


Results - Two Agent

The **two-agent** problem (both agents had full information(s^{env}), **proved to be too complex for any of the algorithms** we implemented on the single agent. We started with actor-critic and DDQN[9], but saw almost no learning. We then proceeded to try:

- **Intrinsic rewards and intrinsic fear** [10][11] which essentially try to reward synergistic behaviors
- We also looked into **further shaping the reward** even specifying certain object locations or relative distances to encourage cooperation, but had no success





Conclusions

- **DQN** showed inconsistent performance, and the lack of convergence resulted in being **unsuccessful** to our goal criterion
- **DDQN and Actor Critic both proved to be more stable methods that achieved success**, with DDQN **performing the best of the 4 algorithms achieving 200s on every angle between -12 and 12 degrees but** would experience bouts of catastrophic forgetting.
- **Rainbow was considered successful** as well, as it learned fastest in term of episodes but had a slightly lower average than the DDQN agent.
- **The two-agent problem proved to be too complex**, with more research needed on encouraging efficient search of the state space for synergistic behavior. **We believe more research is needed in a combination of reward shaping and multi-agent intrinsic reward algorithms combined with significantly more training.** This might result in reasonable performance on the multi-agent version.

Citations

[1] Kumar, Swagat. 2020. "Balancing a CartPole System with Reinforcement Learning -- A Tutorial." *ArXiv:2006.04938 [Cs]*, June
[2] Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press.
[3] Gustafsson, Fredrik. n.d. "Control of Inverted Double Pendulum Using Reinforcement Learning".
[4] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. n.d. "Playing Atari with Deep Reinforcement Learning," 9.
[5] Hasselt, Hado van, Arthur Guez, and David Silver. 2015. "Deep Reinforcement Learning with Double Q-Learning." *ArXiv:1509.06461 [Cs]*, December.
[6] Choi, Minsuk. 2019. "An Empirical Study on the Optimal Batch Size for the Deep Q-Network." In *Robot Intelligence Technology and Applications 5*, edited by Jong-Hwan Kim, Hyun Myung, Junmo Kim, Weiliang Xu, Eric T Matson, Jin-Woo Jung, and Han-Lim Choi, 73–81. Cham: Springer International Publishing.
[7] Hessel, Matteo, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2017. "Rainbow: Combining Improvements in Deep Reinforcement Learning."*ArXiv:1710.02298 [Cs]*, October.
[8] Lowe, Ryan, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2020. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments." *ArXiv:1706.02275 [Cs]*, March.
[9] Hafiz, Abdul Mueed, and Ghulam Mohiuddin Bhat. n.d. "Deep Q-Network Based Multi-Agent Reinforcement Learning with Binary Action Agents," 14.
[10] Chitnis, Rohan, Shubham Tulsiani, Saurabh Gupta, and Abhinav Gupta. 2020. "INTRINSIC MOTIVATION FOR ENCOURAGING SYNERGISTIC BEHAVIOR," 15.
[11] Lipton, Zachary C., Kamyar Azizzadenesheli, Abhishek Kumar, Lihong Li, Jianfeng Gao, and Li Deng. 2018. "Combating Reinforcement Learning's Sisyphean Curse with Intrinsic Fear." *ArXiv:1611.01211 [Cs, Stat]*, March.
[12] Hu, Yujing, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. 2020. "Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping." *ArXiv:2011.02669 [Cs]*, November.