ID 19) Requirement Report: Combining Behavioral Cloning and Reinforcement Learning

Overview

Initially, we theorized that combining Behavioral Cloning and Reinforcement Learning within tasks would be a way for us to generate more accurate results for our tasks. However, upon further examination, the benefits are not as apparent for the size of model we are currently training. There are two methods for which we could consider combining algorithms, one is to use an integrated approach while the second is to add loss functions for well-trained models.

Integration of Behavioral Cloning and Reinforcement Learning is an open area of study within the field. Goecks et al [1] describe an approach they developed which they coined a cycle-of-learning framework. This method involves using an off-policy, actor critic architecture to pre-train a policy and a value function using demonstrations and Behavioral Cloning. The information obtained from Behavioral Cloning is stored in a series of state-action pairs that will cyclicly be drawn on by the agent to allow for Behavioral Cloning to continue to shape the agent's behavior. While the authors provide pseudocode for their approach, it is unclear how we would map this approach onto our task. Additionally, we would likely have to restructure the way we are performing our Behavioral Cloning to generate both the policy and a value function. While such a task would be exciting to accomplish, it would take a great amount of research and model reconfiguration, a task which is primarily performed by PhD researchers. The task would thus need to be restructured into multiple segments to understand the process and learn how to implement it.

The second, more straightforward approach would be to simply combine loss functions generated individually by both models. While this approach has been documented as utilized in practice, it is not easily converted to our specific task. Rajeswaran et al. [4] used an approach that combined loss functions to train an AI hand to perform tasks such as hammering and opening a door. Their approach was to combine the aforementioned functions in such a way as to achieve the maximum benefit of both methodologies. Behavioral Cloning excelled in training the agent to perform specific tasks, but resulted in jerky movement patterns. Their approach was to combine this loss function with one constructed through Reinforcement Learning to smooth out behaviors. However, such a methodology does not map cleanly to our goals, as we are not currently encountering problems with transitions and jerky movements, rather difficulties getting a model to perform as desired in the first place. Additionally, this process would require us to have a well-trained Reinforcement Learning model. As mentioned in our previous sprint, this is not a feasible task for us to accomplish for the task of chopping trees. Rewards are

provided too infrequently within the environment for adequate training of the model. Other tasks and environments that more frequently reward the AI would theoretically be possible to adequately train a Reinforcement Learning model with. Such a task could be stone collecting, as stone is a much more readily available resource than wood, and its collection methodology is significantly simpler, since digging down would reward the AI with accomplishing this task.

Future Development and Training Direction

Imitation Learning: [3] One of the main approaches mentioned and demonstrated in the original MineRL documentation is Imitation Learning. This approach periodically petitions humans to weigh in on the AI's behavior during training to help determine which course of action is better for the AI to be engaging with. While initially we thought we wouldn't need to pursue this path, it would probably be the most directly implementable method for trying to train the AI on other tasks.

Direct Policy Learning (via Interactive Demonstrator) [2] This method builds upon Behavioral Cloning and Imitation Learning by (1) training a policy off of demonstrations, (2) running the policy in the environment, (3) querying an human observer about the results, (4) feeding the human feedback back into the model for continued training. This process repeats steps 3 and 4 until the loop converges, and no additional data can be garnered (or until stopped by the human operator). Conceptually, this seems like a good direction to pursue; however, after recent shortcomings, it might be best to pursue further research as to its complexity and hardware requirements before committing to such an approach.

[1] Goecks et al., "Integrating Behavior Cloning and Reinforcement Learning for Improved Performance in Dense and Sparse Rewards Environments" arXiv.org, https://www.ifaamas.org/Proceedings/aamas2020/pdfs/p465.pdf (accessed Nov 3rd, 2023).

[2] Zoltan Lorincz., "A brief overview of Imitation Learning" Medium.com https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c (accessed Nov 4th, 2023)

[3] Shah et al., “NeurIPS 2021 Competition proposal: The MineRL BASALT Competition on Learning from Human Feedback” arxiv.org, https://arxiv.org/abs/2107.01969 (accessed Nov 5th, 2023)

[4] Rajeswaran et al., “Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations” roboticsproceedings.org https://www.roboticsproceedings.org/rss14/p49.pdf (accessed Nov 5th, 2023)