Coping with sample inefficiency of deep-reinforcement learning (DRL) for embodied AI

WiML Un-Workshop @ ICML 2020

July 13, 2020

Session Leaders: Vidhi Jain, Simin Liu
Session Facilitators: Ganesh Iyer

1 Introduction

Learning has demonstrated great potential for applications like robotics, self-driving cars and IoT [1]. Many of its notable successes have been in virtual gameplay, where thousands of hours of training is feasible. However, collecting real-world data is laborious. To this end, we discuss ways to make DRL methods more practical for embodied AI. Deep reinforcement learning has had great success in simulation, for example, AlphaGo [2] beat human experts, Deepmind's AlphaStar [] beat top professional players at StarCraft, a challenging real-time strategy game, in 2019.. Similarly, OpenAI Five's DOTA bot [] won the championship. But it has been much harder applying it to real world platforms (like robots, autonomous vehicles, process control).

There are many reasons why robotic learning is challenging in real world - lack of safety assurances, reward specification, lack of progress on the continual learning front. One of the major focus is mainly on lack of sample efficiency. One may consider it is not an issue if we have good simulators as we can get loads of training data. But it seems a big issue is for real platforms where acquiring data is effortful, time consuming, and expensive.

We should look into real embodied AI learning in order to fundamentally enhance the notion of intelligence by incorporating multi-modal interaction. More importantly, we want to be efficient and reduce monotonous burden with AI tools like autonomous vehicle and home assistants.

In our breakout session, we discussed about some ways (either proven or promising) to make DRL feasible for embodied AI. We talked about two divergent approaches: algorithmic approaches to improve sample efficiency and alternatively, circumventing the sample efficiency problem by scaling up data collection for current state-of-the-art algorithms.

2 Choice of paradigm

2.1 Model-based or model-free?

Model-based is more sample efficient. However, the big caveat is that its asymptotic performance levels out early (unlike model-free).

2.2 Off-policy or on-policy?

Off-policy is more sample efficient. Off-policy updates have been shown to learn policies for robotics manipulation [3] in a sample efficient way.

3 Using knowledge from other domains

Transfer learning is a kind of "Forward transfer". Transfer learning refers to when a model trained on one task data can be applied or transferred to a new task. from mid level visual feature representations has shown effective success in embodied AI navigation tasks. Similarly, transfer approaches have been applied in language based navigation and modular approach for vision-language grounded navigation tasks.

Multi-task learning implies that the policy is trained to solve many tasks, and show good generalization to an unseen task. This approach is inspired from how human babies learn not a single but many different tasks, and has success in reinforcement learning, particularly evolutionary policy search. DeepMind's UNREAL agent learns to navigate by training on auxillary tasks in an unsupervised manner.

Meta learning refers to learning to learn by solving a variety of tasks. The meta policy learns to minimize the distribution of loss functions that allows it to generalize well to few-shot learning settings. Meta learning has been shown to enable adaptability in legged robots in case of leg joint failures.

Combining modular components of learning systems for embodied AI has been successful approach in Active Neural SLAM and object navigation challenges recently where the policy is decomposed into a mapper, global and local policy. These components have been shown to work on real LocoBot using MaskRCNN for object detection and segmentation.

4 Human Demonstrations and feedback

Copying experts reduce the number of samples needed to explore the state-action space for the optimal policy. Vanilla imitation learning involves extraction of state-action pairs from collected expert demos (trajectories), and minimization of error between policies and next actions. [4] Incorporating demonstrations can also be investigated with learning the Q-value and through inverse reinforcement learning.

5 Scaling up data collection

5.1 Sim-2-real

Can we train primarily in simulation and then transfer the policy to the real world? Sim-2-real approach allows for cheap data collection, effortless scaling and safe setting to learn policies. But we need to deal with the 'sim-2-real gap'. There are often significant visual and physical differences between the simulation and reality.

The simplest transfer case where we can naively go from simulations to real-world, requires that we have a simulator with the perfect model of the world. Domain adaptation and domain randomization have shown empirical success with neural networks. Domain adaptation refers to approach of finding a robust mapping from simulation to reality. Domain randomization refers to perturbing the simulation to different extents in the hope that generalization to reality becomes an interpolation task (rather than a difficult extrapolation one).

5.2 Parallelized methods

5.2.1 Parallelized asynchronous data collection

edge workers merely send raw data to the server for training. Current attempts to parallelize data collection (Arm Farm at Google), algorithmic changes need to be made to suit in asynchronous, parallel data collection (i.e. straggler mitigation) For embodied AI navigation, [5] scaled the data collection and execution in decentralized way, while maintaining the centralized synchronized training updates.

5.2.2 Federated learning

Edge workers update personal models and asynchronously send model parameters to update the global features on the server. Federated learning allows the robot to learn local features for adaptation, leverage the personal data in privacy preserving way, and communicate the model parameters efficiently for global feature representation. Federated learning approach has been successful in improving on-device voice assistants for keyword spotting [6], and is an open research area for self-driving cars and IoT [7]. Federated deep reinforcement learning is an active area of research [8].

6 Miscellaneous

Embedding strong inductive priors. For example, reward shaping [9], crafting exploratory behavior [10], and incorporating human feedback or demonstrations [11]. Additional topics include algorithmic changes to make more efficient use of data, like experience replay [11] or curriculum learning.

7 Discussion

- When is DRL useful/necessary for embodied AI applications? (i.e. when do data-driven methods have an advantage over traditional planning control methods?)
- Is sample inefficiency a bottleneck in the progress of DRL for robotics? Pro: Sample inefficiency deters real-world applications, adaptation of the algorithms Cons: Sample inefficiency in current methods can be overcomed with improved faster hardware and realistic simulation. Though sample inefficient, the current algorithms are somewhat self-reliant - perhaps we have a tradeoff here.
- Why is the sample inefficiency of current DRL algorithms not so serious? - improved hardware, simulation power, less reliance on known priors and more on rules extracted from patterns in the data i.e. data-driven?
- Does Sim2Real work? Just by publishing papers on the success of sim2real, we can not believe that sim2real works. There are no negative examples that we can discuss! Since sim-2-real somewhat works, can't we just use any of our DRL algorithms, even if data inefficient?
- Should our focus as a community be on circumventing sample efficiency (gathering data at scale, sim-2-real) or address it at algorithmic level? Should we modify our learning algorithm or not? (focus: embodied AI learning models)
- Do you see ways in which these methods can be combined? What's wrong with the way we currently measure/quantify sample efficiency?

8 Open Questions

- Should we expend more effort in improving the realism of our simulations or on algorithms to compensate for their lack of realism? If the first: how do we measure simulation realism? If the second: how do we make algorithms less ad hoc (arbitrary) than they are currently (esp. Domain Randomization - isn't the way you permute is arbitrary)?
- Federated learning has shown early promise in areas like query suggestions on mobile phones, smart speakers, etc. What other applications can you think of?
- What kind of realism is desirable - like visual realism through meshes or physical realism through CAD models and physics based game engines? For example, game engines like Unity (AI2THOR) can provide similar physical realism. Real world 3D mesh renderers like AI Habitat provide with visual realism in simulators.

References

[1] Lei Lei, Yue Tan, Shiwen Liu, Kan Zheng, and Xuemin Shen. Deep reinforcement learning for autonomous internet of things: Model, applications and challenges. *CoRR*, abs/1907.09059, 2019.

[2] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[3] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.

[5] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames, 2019.

[6] Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning, 2020.

[7] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. Federated learning for resource-constrained iot devices: Panoramas and state-of-the-art, 2020.

[8] Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. Federated deep reinforcement learning, 2019.

[9] Glen Berseth, Daniel Geng, Coline Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise minimizing reinforcement learning in dynamic environments, 2019.

[10] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 488–489, 2017.

[11] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5048–5058. Curran Associates, Inc., 2017.