Guaranteeing Reproducibility in Deep Learning Competitions

Brandon Houghton *
Carnegie Mellon University

Stephanie Milani
Carnegie Mellon University

Nicholay Topin
Carnegie Mellon University

William Guss
Carnegie Mellon University

Katja Hofmann
Microsoft Research

Diego Perez-Liebana
Queen Mary Univ. College of London

Manuela Veloso
Carnegie Mellon University

Ruslan Salakhutdinov
Carnegie Mellon University

1 Introduction

Democratizing access to artificial intelligence (AI) requires competitions that promote the development of sample-efficient learning, as well as ensure the reproducibility and generalizability of results. Sample efficiency is important because practitioners with limited compute resources cannot readily utilize algorithms that require a massive number of samples. The complexity of these state-of-the-art methods is outpacing advancements in computation. Moreover, as methods and domains become more specialized, learning procedures become more fragile: often undocumented modifications can inhibit reproducible results and seeds are chosen to reflect the optimal performance of a given solution [Henderson et al., 2018].

Because the focus of traditional research challenges is the development of new techniques in a particular field, these challenges seek to reward participants for novel solutions. However, submissions with the best performance on the (often highly specified) task tend leverage domain knowledge that is not broadly applicable, leading challenges to open separate tracks where submissions are subjectively evaluated on research novelty [Pavlov et al., 2018].

To encourage participants to develop methods with reproducible and robust training behavior, we propose a challenge paradigm where competitors are evaluated directly on the performance of their learning procedures rather than pre-trained agents. Since competition organizers re-train submissions in a controlled setting they can guarantee reproducibility, and – by retraining submissions using a held-out test set – help ensure generalization of submissions past the environments on which they were trained.

2 Case Study: MineRL

We use the aforementioned paradigm in our competition, the MineRL Competition on Sample Efficient Reinforcement Learning [Guss et al., 2019]. Through this competition, we challenge the deep reinforcement learning (DRL) community to train an agent to solve a complex, hierarchical task using limited computation time and a fixed budget of 8 million environment samples. To assist with the development of their algorithms, participants can leverage a large annotated dataset of demonstrations [Guss* et al., 2019] through a competition starter kit [MineRL, 2019]. To ensure that winning entries can be reproduced, organizers retrain submissions in the final round using an entirely new, previously-unseen texture pack [Wiki]. Because the competition organizers supervise the training

*bhoughton@cmu.edu

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

procedure, they can ensure that submissions hold to specific constraints (such as using a limited amount of environment samples and training time). This requirement also prevents participants from using prior knowledge of the environment, Minecraft, to hand-craft policies.

2.1 Computational Requirements

An important concern is the additional budget that this evaluation structure requires of the organizers. In particular, organizers need additional computational power to retrain participant models. In order to reduce the computation required when re-training competitor submissions in the MineRL competition we chose to limit re-training to round two, where the top ten teams from round one compete. This structure allows us to open the competition to any number of interested participants while constraining the computational budget. Additionally, by inviting only the top ten teams to a second round, we can provide each team with up to five attempts to re-train their model in round two. We encourage organizers of future competitions to consider a similar scheme for computation allocation as a way to provide a sufficient amount of computational resources to top teams while simultaneously not limiting the number of competition participants.

2.2 Data Requirements

Requiring algorithms that restrict compute time and number of environment samples can result in solutions which underfit to the training data or that fail to learn even simple tasks. One way to improve the sample efficiency of learning algorithms is to use demonstrations [Dubey et al., 2018]. Many widely-used imitation learning methods require the label of the demonstration to be provided by an expert [Ho and Ermon, 2016]; however, expert labelling is generally prohibitively expensive. As an alternative, defining simple metrics (such as time to completion) for sub-tasks that we believed would be useful for competition, allowed us to crowd source demonstrations and provide the competitors the option to sample both expert and non-expert trajectories.

In addition to computational requirements, organizers need enough data so that both the original training set and held-out set are sufficiently large. To meet this requirement, we recorded our dataset in such a way that it can be easily re-rendered and altered. Specifically, to create a new dataset using the original recordings, we use these recordings to re-simulate the game actions in an environment with visual changes and capture the resulting video stream. As a result, we create two datasets which contain the same higher-level information but which are visually distinct. Through this process, we create two different datasets without reducing the size of either one.

3 Takeaways

Through our competition, we have learned the following lessons that we would like to share with the broader community. First, adding the constraint that all final submissions of the participants are retrained on a new texture of the environment guarantees that their submissions are reproducible and robust to perturbations. This requirement also limits the exploitation of domain knowledge of the environment. Second, if organizers cannot collect a dataset rich enough for standard RL methods, they should consider potential subtasks that could be learned in a simpler domain. Data on these auxiliary tasks can then be given to participants to help their agents learn more basic, composable skills. Third, some participants may become disengaged when issues impede their submission's development. Providing both text and video demonstration of submission gives participants confidence that their idea can be executed and encourages them to continue developing their submission.

4 Conclusion

We propose a novel challenge paradigm in which competitors are (1) evaluated solely on the performance of their learning procedures instead of on pre-trained agents and (2) encouraged to produce learning algorithms which prioritize sample efficiency. Through the case study of the MineRL competition, we show that this paradigm is possible and provide examples of how to implement this paradigm in practice. Although the proposed paradigm is computationally expensive for the competition organizers, enabling research competitions in machine learning to yield reproducible and sample-efficient methodologies provides great benefit to the community.

Acknowledgments

We thank the following people for their contributions to the MineRL competition: Cayden Codel, Phillip Wang, Noboru (Sean) Kuno, Sharada Mohanty and AICrowd, Shivam Khandelwal, Crissman Loomis, Nakata, Shohe Hido, and Preferred Networks, Harm van Seijen, Mario Ynocente Castro, Shinya Shiroshita, Andre Kramer, Chelsea Finn, Oriol Vinyals, and Sergey Levine.

References

Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L. Griffins, and Alexei A. Efros. Investigating human priors for playing video games. In *ICML*, 2018.

William H. Guss, Cayden Codel*, Katja Hofmann*, Brandon Houghton*, Noboru Kuno*, Stephanie Milani*, Sharada Mohanty*, Diego Perez Liebana*, Ruslan Salakhutdinov*, Nicholay Topin*, Manuela Veloso*, and Philip Wang*. The MineRL competition on sample efficient reinforcement learning using human priors. In *NeurIPS Competition Track*, 2019.

William H. Guss*, Brandon Houghton*, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. MineRL: A large-scale dataset of Minecraft demonstrations. In *IJCAI*, 2019.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016.

MineRL. MineRL competition submission starter template, 2019. URL `https://github.com/minerllabs/competition_submission_starter_template`.

Mikhail Pavlov, Sergey Kolesnikov, and Sergey M. Plis. Run, skeleton, run: Skeletal model in a physics-based simulation, 2018. URL `https://arxiv.org/abs/1711.06922`.

Minecraft Wiki. Minecraft wiki. URL `https://minecraft.gamepedia.com/Texture_pack`.