Enhance Deep Reinforcement Learning using Curriculum Learning

October 2020

In this project, we aim to learn two broad topics in Machine Learning, which have recently found many applications but their core ideas are rooted way past in the history of mathematics. In particular, we will study and discuss Reinforcement Learning (RL) and Curriculum Learning.

Reinforcement Learning is quite similar to how humans learn: when learning a new task without a teacher, we map situations to actions—to maximize a numerical reward signal [Forestier et al., 2020]. The learner is not told which actions to take but instead must discover which actions yield the most reward by a hit-and-trial approach. In addition to this simple scenario, if we consider the game of chess, a player's actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. Trial-and-error search and delayed reward—are the two most important characteristics of reinforcement learning. Unlike supervised learning where we map features to prediction class using a passive form of learning, RL finds a unique approach to learning actively.

The ancient game of Go has been a challenge in AI for decades and has been the subject of much recent research: AlphaGo and its variants can now reliably beat the world's best human professional Go players [Wang et al., 2019]. Each such achievement represents the pinnacle of a long march of increasing performance and sophistication aimed at solving the problem at hand. Deep reinforcement learning algorithms have been used to solve difficult tasks in video games, locomotion, self-driving, and robotics. As compelling as this narrative may be, it is not the only conceivable path forward. Tasks with sparse rewards like "Robot, fetch me a beer" remain challenging to solve with the direct application of these algorithms [Matiisen et al., 2017]. One reason is that the number of samples needed to solve a task with random exploration increases exponentially with the number of steps to get a reward [Langford, 2010]. Curriculum learning [Bengio et al., 2009] may be used to directly tackle such problems, where tasks are ordered by increasing difficulty and training only proceeds to harder tasks once easier ones are mastered. Curriculum learning helps when after mastering a simpler task the policy for a harder task is discoverable through random exploration. Prime challenges for including Curriculum while training is: (1) derivation of subtasks given the final tasks [Bengio et al., 2009]. (2) Continuously mix easier tasks so that one can avoid forgetting easier tasks[Matiisen et al., 2017].

[Matiisen et al., 2017] addressed the above-mentioned challenges of automatic curriculum learning to some extent and presented a framework of Teacher-Student Curriculum Learning (TSCL), where the Student (agent) tries to learn a complex task and the Teacher automatically chooses subtasks from a given set for the Student to train on. This unique component (Teacher network) enabled them to solve a Minecraft maze that could not be solved at all when training directly (with traditional RL) on solving the maze, and the learning was an order of magnitude faster than a uniform sampling of subtasks. In this project, we will extend their work. Our RL tool will include a curriculum component in the form of a TSCL network with additional improvements. We list some limitations we noticed in their work. First, they assume all the tasks are in the same domain(game), which ignores the opportunity to learn from any other domains (games). Second, the authors do not consider the model capacity w.r.t to the difficulty of the tasks which means the model will adjust the structure or the number of parameters of the model based on the tasks' difficulty. Third, all the experiments shown have a limited number of subtasks, we would like to see if the original hypothesis changes if there are thousands of subtasks. Fourth, they have a hyperparameter for the number of steps a student should train on each sub-task, without any empirical or analytical evidence. Finally, these subtasks are pre-defined by experts, however these subtasks can also be generated from a generative model i.e it can automatically generate the optimal environment for the student to learn, this will also enable the student to learn more complex tasks indefinitely.

Accordingly some of the possible contributions as a part of our tool are:

- **Introduce Model curriculum in TSCL, for example Prgressive GANs [Karras et al., 2017] where one has to increase the model capacity as the difficulty of the subtasks increase.**

- **Apply TSCL to CoinRun game which has 1000 subtasks. This will provide more robustness to our experiments.**

- **Enhance the migration from one task to another by deriving more task related metrics that can automatically determine how dissimilar are two sub tasks.**

The authors have provided the TSCL code on GitHub [Link, b]. We plan to extend it with our contributions. We are planning to modify the implementation to make it more compute efficient, for example, we can simplify the PacMan environment. As per the tools, we plan to use Open AI Gym and Tensorflow for our project. More details about our code and implementation can be found in our GitHub Repo [Link, a].

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

John Langford. *Efficient Exploration in Reinforcement Learning*, pages 309–311. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_244. URL https://doi.org/10.1007/978-0-387-30164-8_244.

Code Link, a. URL https://github.com/harsh306/rl-project.

TSCL Code Link, b. URL https://github.com/tambetm/TSCL.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning, 2017.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions, 2019.