Large Language Models in Reinforcement Learning - A Comparison of Methods

Adrian Duric
Master Student, Dept. of Informatics
The Faculty of Mathematics
Oslo, Norway
adriandu@ifi.uio.no

Gregor Kajda
Master Student, Dept. of Informatics
The Faculty of Mathematics
Oslo, Norway
grzegork@ifi.uio.no

Jonatan Hoffmann Hanssen
Master Student, Dept. of Informatics
The Faculty of Mathematics
Oslo, Norway
jonatahh@ifi.uio.no

***Abstract*—Reinforcement Learning (RL) algorithms suffer when rewards are sparce and the state-action space is large. Even tasks which appear relatively simple can prove intractable if the completion of the task requires subtasks to be completed first, or if the reward only comes when the entire task has been completed. In such cases, random exploration is unlikely to lead the agent to discover a solution to the problem. The apparent simplicity of such problems is often due to human intuition, which allow us to quickly see possible solutions to a diverse set of problems. Large Language Models (LLMs) are trained on large corpura of human written text, and have been shown to capture parts of this intuition in many tasks [Bubeck et al., 2023]. In recent years, LLMs have been succesfully used to aid RL agents in more efficient exploration, and have allowed them to solve problems which previous methods have been unable to [Zhang et al., 2023] [Du et al., 2023]. We compare different methods for integrating an LLM into the deep learning reinforcement algorithm proximal policy optimization (PPO), and compare their efficiency against eachother. We find that BEST METHOD gives the best sample efficiency, outperforming normal PPO by PERCENTAGE in METRIC.**

***Index Terms*—large language models, reinforcement learning, minigrid, proximal policy optimization**

I. Introduction

One of the central challenges of reinforcement learning is that rewards are often both extremely rare and delayed, which means that optimizing a reinforcement learning algorithm requires significant trial and error [Brunton and Kutz, 2022, 423]. Furthermore, in many problems the state-action spaces are enormous, meaning that uniform exploration is unlikely to find good solutions in a reasonable amount of time. Many methods have been developed to deal with this issue, for example by introducing auxilliary reward functions that use domain knowledge to reward actions which are considered good, or which reward the agent for learning novel skills. However, hand picking which actions to reward can lead to imitation rather than optimal behaviour, and novelty is not always useful [Du et al., 2023, 1]. In recent years, LLMs have shown remarkable capabilities in problem solving and planning [Bubeck et al., 2023], qualities which traditional RL agents often lack. Thus, a new area of research has emerged, which attemps to use the vast amount of human knowledge encoded in these models to increase the performance of reinforcement learning algorithms [Luketina et al., 2019]. These LLM assisted RL agents have been able to outperform state-of-the-art RL methods in many problems [Zhang et al., 2023] [Du et al., 2023] [Li et al., 2022].

These papers explore different methods of integrating LLMs into a standard reinforcement learning training loop, from altering the policy directly to introducing an auxilliary reward function. In this paper, we compare both methods in the same environment using the same deep reinforcement algorithm (PPO), and compare their results to eachother to explore how LLMs best can be integrated into RL.

II. Background and Related Work

A. Large Language Models as Policy in Reinforcement Learning

In recent research into integrating LLMs into the RL framework, many of the methods proposed include ways of making the LLM influence the policy of the RL agent. [Zhang et al., 2023] considers open-ended learning algorithms in which the value of the agent attempting certain tasks can be evaluated by estimating the the interestingness of the task. The notion of interestingness is then thought of as what task would intuitively be interesting for a human to try in the context of learning in an environment. Some main factors contributing to this interestingness are how likely one is to succeed in doing the task, as well as whether learning the given task may increase the likelihood of succeeding in other tasks. Considering that LLMs are trained on extremely large amounts of human-written text containing human knowledge and intuition, this leads to the idea that if the LLM was told to propose interesting actions for the RL agent to learn in its environment, it could be able to communicate human intuition directly to the agent. This would in turn let the agent learn with increased sample efficiency as the human intuition-based input it receives would make it decide to perform intuitively smarter actions, particularly in the early stages of exploring its environment.

The algorithm proposed in this paper involves prompting the LLM with the agent's state observation, a list of tasks the agent already does well, and a list of available tasks for the LLM to deem as either interesting or boring. After sorting available tasks into interesting and boring, it assigns sampling

weights to them, giving much higher weights to interesting tasks than boring ones, thus directly influencing the agent's policy by altering probabilities of which action it then chooses to do. In [Li et al., 2022], another paper proposing using the LLM as a policy optimizer, has a similar approach in which the state observation is tokenized and passed to the LLM, though in their experiment, goals and action histories are also tokenized and passed. Here, too, the LLM response is fed back into a task-specific RL model, which in turn influences policy probabilities of certain actions being taken. In general, many of the novel methods suggested involve the LLM suggesting actions for the agent to perform in a given context, and its output influencing the agent policy.

B. Large Language Models as Reward in Reinforcement Learning

Another often proposed method of involving LLMs in RL, is to have it influence the reward the agent receives in certain states. In particular, [Du et al., 2023] cites the infrequency of rewards as a common bottleneck in RL algorithms, due to how long exploratory trajectories often have to be before the agent reaches some desirable state. This holds true especially for large, complex environments, where the probability of reaching favorable states among a huge number of non-favorable states becomes even smaller. Similarly to the papers proposing LLMs to influence policy, this paper too seeks to make use of human intuition and knowledge present in the vast training data that state-of-the-art LLMs have been trained upon, to make the agent prioritize exploring plausibly useful behaviors first, based on some intelligent forethought provided by the LLM.

However, rather than doing so through directly influencing action probabilities in the policy, the paper suggests rewarding similarity to goal states described by the LLM. At certain timesteps, the LLM would then be prompted with a state observation from the agent. It would be told to suggest goal states for the agent to reach. Some timesteps later, if the new state observation is semantically similar to the previous LLM output, this would give a reward to the agent. Thus, the proposed algorithm makes use of LLMs in two ways: first to generate what the LLM perceives as desirable states, then later to assess similarity between a natural-text description of the current state and the previously described goal state. The first use case in particular utilizes the presence of human intuition in LLMs, as a human too would often be able to intuitively figure out what is desirable to achieve in some environment.

III. Methods

A. Problem Formulation

We train a PPO model in a Partially Observable Markov Decision Process (POMDP). A POMDP is defined by a tuple $(S, A, T, R, O, \Omega, \gamma)$. Here, $s \in S$ denotes the state the environment is in and $a \in A$ denotes an action the agent can take. Based on the state and the action taken, the agent receives an observation $o \in O$ which depends on the new environment state and the action taken by $O(o|s, a)$. $T(s'|s, a)$ denotes the state transfer function, which gives a new state based on the previous state and action. The environment we use is part of Minigrid [Chevalier-Boisvert et al., 2023].

Deguser trenger god plass for å trives, og du bør investere i et størst mulig bur som gir dyrene mulighet til å klatre, grave i et tykt strølag og bevege seg i flere etasjer.

Mange dyreeiere konstruerer sine egne løsninger, for eksempel med et stort, gammelt akvarium som underdel og et nettingbur på toppen av dette. Da får du full oversikt over hva som skjer også nederst i buret, samtidig som du slipper at sand, strø og høy søles ut i rommet. Ikke la degusene gå direkte på gitterbunn eller annet hardt underlag, det kan skade føttene.

Buret kan du innrede med f.eks. røtter og greiner til å klatre på, hus av treverk, rør til å krype gjennom og løpehjul. Et løpehjul til degus bør være minst 25 cm i diameter og ha tett gulv og vegger slik at halen ikke kan komme i klem. Bunnmaterialet i buret bør være et tykt lag av støvfritt smådyrstrø som er egnet for graving. Det finnes flere egnede strøtyper å få kjøpt. Bland gjerne inn litt tørr blomsterjord eller sand i bunnmaterialet. Hvis du henter sand ute, bør du desinfisere den ved frysing eller steking fr den brukes i buret. Degusene trenger også tilgang på fiberrikt materiale som høy, revet papir eller treull til bygge- og redemateriale.

For å være sikker på at alle degusene som bor sammen kan trekke seg unna og hvile, bør de få hvert sitt sovehus. Deguser kan også finne på å forsvare maten sin overfor andre deguser. Når flere deguser bor i samme bur, bør du derfor ha flere matskåler plassert på forskjellige steder i buret slik at alle får tilgang til mat når de ønsker det. Dyrene må alltid ha tilgang på friskt vann, som best gis i en drikkeflaske.

For å holde pelsen i orden har deguser behov for å bade i finkornet sand flere ganger i uka. Slik badesand får du kjøpt i dyrebutikken. Et tungt kar som ikke velter så lett er fint som sandbadekar. Det kan være lurt å fjerne sandbadet mellom hvert bad for å unngå at den brukes som toalett.

Rengjøring

Hvor ofte buret bør rengjøres avhenger av størrelsen og hvor mange som bor der. Som en tommelfingerregel bør du skifte bunnmaterialet og vaske bunn og innredning én gang i uka.

IV. Experiments

V. Results

VI. Conclusion

Acknowledgment

I would like to acknowledge myself for being a absolute legend.

References

[Brunton and Kutz, 2022] Brunton, S. L. and Kutz, J. N. (2022). *Data-Driven Science and Engineering*. Cambridge Univeristy Press, Cambridge, UK.

[Bubeck et al., 2023] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4.

[Chevalier-Boisvert et al., 2023] Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S., and Terry, J. (2023). Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831.

[Du et al., 2023] Du, Y., Watkins, O., Wang, Z., Colas, C., Darrell, T., Abbeel, P., Gupta, A., and Andreas, J. (2023). Guiding pretraining in reinforcement learning with large language models.

[Li et al., 2022] Li, S., Puig, X., Paxton, C., Du, Y., Wang, C., Fan, L., Chen, T., Huang, D.-A., Akyürek, E., Anandkumar, A., Andreas, J., Mordatch, I., Torralba, A., and Zhu, Y. (2022). Pre-trained language models for interactive decision-making. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31199–31212. Curran Associates, Inc.

[Luketina et al., 2019] Luketina, J., Nardelli, N., Farquhar, G., Foerster, J., Andreas, J., Grefenstette, E., Whiteson, S., and Rocktäschel, T. (2019). A survey of reinforcement learning informed by natural language.

[Zhang et al., 2023] Zhang, J., Lehman, J., Stanley, K., and Clune, J. (2023). Omni: Open-endedness via models of human notions of interestingness.