Documentation of Bit Sequence mini RL project

Chau Yu Hei[*]
HKUST
20644747
yhchau@connect.ust.hk

Abstract

GFlowNets provide a new paradigm for finetuning LLMs with respect to some energy function. However, a common problem is that strings with high probability are often very rare compared to the strings with low probability. How do we efficiently tune LLMs using the GFlowNet algorithm so that it is able to learn from sparse rewards? In this mini-project, we attempt to tackle the problem of generating a randomly prechosen bit string using old RL methods.

1 Background

Recent work has shown that GFlowNets are able to generate diverse samples for LLMs, and finetune LLMs according to Bayesian updates (Hu et al. [2024]). However, their experiments are limited to generating text sequences of short length. How do we generate text sequences of larger lengths? It is evident that most strings have low reward due to the large search space of the set of all possible text sequences.

In this mini-project, we attempt to tackle a similar toy problem, that is, to generate a specific bit sequence in the set of all bit sequences. We first apply a naive DQN method for sequence generation, and show that the method only works for small n. Then we attempt to apply Hindsight Experience Replay (Andrychowicz et al. [2017]) by OpenAI to see whether the method successfully generates the correct bit sequences for longer sequence lengths.

2 Problem setting

Here we attempt to use a DQN to generate a predesignated randomly picked target bit sequence. In each setting, an integer $1 \leq n \leq 50$ is fixed, and the state space is $\mathcal{S}_n = \{0, 1\}^n$. Before the training process, we pre-pick a sequence $s \in \mathcal{S}_n$ randomly (e.g 00110 for $\mathcal{S}_5$). During training, the goal of the DQN network is to generate the sequence s, where the reward is 1 for arriving at s, and 0 for arriving at other $\mathcal{S}_n \ni s' \neq s$.

The intuition is that, for low values of n (maybe $1 \leq n \leq 10$?), a naive DQN is able to learn to arrive at the bit sequence due to exhausting the search space (along with using the parallel processing by GPUs). However, for large values of n, this would be infeasible. We will further explore generating the sequence using other methods.

3 Experiments

Coming soon ASAP......

[*] Currently not a student yet :(

References

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. *Advances in neural information processing systems*, 30, 2017.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. In *Proceedings of the International Conference on Learning Representations*, 2024.