Intersection Crossing using Policy Optimization with Beta Distribution for Continuous Action

Kim Ang Kheang
Graduate School of Science and Technology
Niigata University
Niigata, Japan
f21c501e@mail.cc.niigata-u.ac.jp

Tatsuya Yamazaki
Faculty of Engineering
Niigata University
Niigata, Japan
yamazaki.tatsuya@ie.niigata-u.ac.jp

I. Introduction

Reinforcement Learning (RL) has emerged as a promising technique for autonomous driving applications, enabling agents to make decisions in dynamic environments. Efficiently navigating intersections is a critical challenge in this context, especially when dealing with continuous action spaces. Traditional RL methods often use Gaussian distributions [1], [2] for continuous actions, but they can have limitation.

To address this issue, we propose an innovative extension of the Proximal Policy Optimization (PPO) algorithm [3] that leverages a Beta distribution. The Beta distribution offers more flexibility to model bounded action spaces [4], leading to better exploration and improved performance in intersection crossing tasks. Additionally, we introduce a duel channel neural architecture to accurately predict the α, β values of the Beta distribution, enhancing the agent's ability to sample actions effectively.

The choice of distribution in PPO can significantly impact the algorithm's performance. Utilizing the appropriate distribution, such as Beta for bounded action spaces, can lead to more efficient exploration, quicker convergence, and improved stability during training, whereas the wrong choice, like Gaussian for bounded actions, may result in suboptimal policies and hinder overall learning.

II. Methodology

We utilized the MetaDrive simulator [5]. MetaDrive provides a realistic and dynamic environment for training and testing autonomous driving agent.

A. Actor Network Architecture

The actor network takes a sequential input of 1059 features, including ego state information, environment information, and Lidar cloud point data. It consists of two hidden layers, each containing 256 neurons. The output is then split into two fully connected layers to predict α, β values for acceleration and steering angle, respectively.

B. Mapping to Action Space

As the simulation environment accepts action with $a = [sa, acc]^T = [-1, 1]^2$, the output from Beta's Probability Distribution Function is then mapped with the following equation:

$$sa = 2h(\beta_{sa}, \alpha_{sa}) - 1 \tag{1}$$

$$acc = 2h(\beta_{acc}, \alpha_{acc}) - 1 \tag{2}$$

III. Result and Discussion

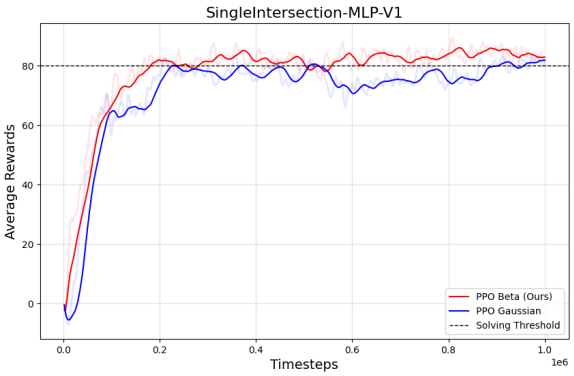

Fig. 1. Beta Policy Network Architecture

We implemented PPO with specific configurations: $\epsilon = 0.02$ (clipping parameter) and $\gamma = 0.99$ (discount factor). The environment solving threshold was set to 80 which means the agent can get through intersection successfully.

PPO Beta achieved the solving threshold faster and maintained higher average rewards throughout training compared to PPO Gaussian. The utilization of the Beta distribution allowed PPO Beta to efficiently explore the continuous action space, leading to faster convergence and higher rewards.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.
[2] A. P. Capasso, P. Maramotti, A. Dell ' Eva, and A. Broggi, "End-to-end intersection handling using multi-agent deep reinforcement learning," 2021.
[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.
[4] I. G. B. Petrazzini and E. A. Antonelo, "Proximal policy optimization with continuous bounded action space via the beta distribution," 2021.
[5] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," 2022.