대형 언어 모델을 활용한 로보틱스 강화학습의 샘플 효율성 향상 연구

박주경, 차채연, 박형곤

이화여자대학교 공과대학 전자전기공학과

{pjk0630, chaeyeon.cha, hyunggon.park}@ewha.ac.kr

Enhancing Sample Efficiency in Robotic Reinforcement Learning Using Large Language Models

Jukyung Park, Chaeyoen Cha, Hyunggon Park

Department of Electronic and Electrical Engineering, Ewha Womans University

요 약

로보틱스와 같이 복잡한 고차원 문제를 다루는 환경에서 오프 폴리시 기반 심층 강화학습 알고리즘은 실제 환경에서 궤적 데이터를 확보하는 데에 높은 시간적, 경제적 비용이 수반되는 한계가 있다. 따라서 본 논문은 대형 언어 모델을 통해 알고리즘의 학습에 필요한 환경 동역학 궤적을 예측하는 방법을 제안한다. 제안하는 방법은 상태 내 구성 요소의 중요도를 반영하는 가중치 행렬을 사용하여 LLM이 보상함수를 모르는 상태에서도 환경 동역학을 예측하여 샘플을 생성한다. Walker2d-v4 환경에서의 실험을 통해 제안 방법이 기존 샘플 생성 방법과 유사한 평균 성능을 가지며 더 빠른 초기 성능 향상을 달성하는 것을 확인하였다.

Ⅰ. 서론

강화학습(Reinforcement Learning, RL)은 에이전트(agent)가 환경과의 상호작용을 통해 최적의 행동 폴리시(policy)을 학습함으로써 순차적 의사결정 문제를 해결하는 방법으로, 자율주행, 로보틱스 등 다양한 분야에서 널리 활용되어 왔다. 특히, 로보틱스에서는 복잡한 고차원 문제를 해결해야 하므로 심층 신경망을 사용하는 심층 강화학습(Deep Reinforcement Learning, DRL)을 활용한 연구가 활발히 진행되어 왔다. 하지만, 오프 폴리시(off-policy) 기반 DRL 알고리즘의 경우, 학습에 필요한 충분한 궤적(trajectory) 데이터를 확보해야 하며, 실제 환경에서의 상호작용에는 높은 시간적, 경제적 비용이 수반되는 한계가 있다 [1].

최근 이러한 한계를 극복하기 위해 대형 언어 모델(Large Language Model, LLM)을 DRL의 보조 학습 모델로 활용하는 시도가 진행되고 있다 [2]. LLM은 방대한 사전지식과 문맥 이해 능력을 바탕으로 환경의 동역학(dynamics)을 예측하거나 시뮬레이션할 수 있으므로 [3], DRL에서 에이전트와 실제 환경과의 상호작용 없이 LLM을 통하여 실제 환경의 동역학에 따른 상태(state) 데이터를 생성할 수 있다.

본 연구에서는 LLM을 통해 오프 폴리시 기반 DRL 알고리즘의 학습에 필요한 환경 동역학 궤적을 예측하는 방법을 제안한다. 특히, 제안하는 방법은 에이전트의 행동 폴리시 학습 과정에 상태가 미치는 영향력을 반영하기 위한 가중 행렬을 정의함으로써, LLM이 에이전트의 보상 함수를 직접 알지 않아도 이를 반영한 환경의 동역학을 예측할 수 있다. Walker2d-v4 환경에서의 실험을 통해 제안한 방법이 적은 실제 환경 샘플을 사용하며 향상된 학습 속도로 최적의 행동 폴리시를 학습하는 것을 확인하였다.

Ⅱ. 본론

2-1. 문제 정의

본 연구는 이족보행 로봇 시뮬레이션을 위한 환경에서 에이전트가 넘어지지 않고 빠르게 전진하는 것을 목표로 한다. 에이전트는 몸통과 두 다리로 구성되며, 양쪽의 발목, 무릎, 허벅지에 총 6개 관절이 있다. 환경은 마르코프 결정 과정(Markov Decision Process, MDP)의 구성 요소인 $< \boldsymbol{S}, \boldsymbol{A}, P, R, \gamma >$로 표현될 수 있다. 여기에서 γ는 감가율(discount factor)을 의미하며, 상태 공간(state space) $\boldsymbol{S}$, 행동 공간(action space) $\boldsymbol{A}$, 상태 전이 함수 P, 보상 함수(reward function) R은 다음과 같이 정의된다.

상태 공간(State Space) 상태 $\boldsymbol{s}_t \in \boldsymbol{S} \subset \mathbb{R}^{17}$는 시간 t에서 에이전트의 높이, 관절 각도, 선속도 및 각속도로 구성된 17차원 실수 벡터로 정의된다. 에이전트의 높이를 z_t, 몸통 각도를 θ_t, 양쪽 다리의 6개 관절 각도를 $\boldsymbol{\phi}_t$, 에이전트의 수평 및 수직 속도를 $\dot{x}_t$, $\dot{z}_t$, 몸통의 각속도를 $\dot{\theta}_t$, 각 관절의 각속도를 $\dot{\boldsymbol{\phi}}_t$라 하면, 상태 $\boldsymbol{s}_t$는 다음과 같이 표현할 수 있다.

$$\boldsymbol{s}_t = [z_t, \theta_t, \boldsymbol{\phi}_t^\top, \dot{x}_t, \dot{z}_t, \dot{\theta}_t, \dot{\boldsymbol{\phi}}_t^\top]^\top$$

행동 공간(Action Space) 행동 $\boldsymbol{a}_t \in \boldsymbol{A} \subset \mathbb{R}^6$은 시간 t에서 에이전트의 6개 관절에 y축을 중심으로 인가되는 토크(torque) 제어로 정의한다. 시간 t에서 i번째 관절의 토크를 $\tau_t^{(i)}$라고 하면, 각 토크의 범위는 $[-1, 1]$이 되고, 행동 $\boldsymbol{a}_t$은 다음과 같이 표현할 수 있다.

$$\boldsymbol{a}_t = [\tau_t^{(1)}, \cdots, \tau_t^{(6)}]^\top, \quad \tau_t^{(i)} \in [-1, 1]$$

시간 t에서의 현재 상태 $\boldsymbol{s}_t$와 행동 $\boldsymbol{a}_t$을 기반으로 다음 상태 $\boldsymbol{s}_{t+1}$는 상태 전이 함수 $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$에 의해 결정된다.

보상 함수(Reward Function) 보상 r_t는 시간 t에서 상태 $\boldsymbol{s}_t$와 행동 $\boldsymbol{a}_t$을 기반으로 보상 함수 $R(\boldsymbol{s}_t, \boldsymbol{a}_t)$에 의해 결정된다. 에이전트는 안정적인 자세로 넘어지지 않고 빠르게 걷는 것을 목표로 한다. 에이전트의 직립 보행 보상을 h_t라 할 때, h_t는 다음과 같이 정의된다.

$$h_t = w_h \cdot \mathbf{1}(z_t \in [z_{\min}, z_{\max}] \wedge \theta_t \in [\theta_{\min}, \theta_{\max}])$$

이때, $\mathbf{1}$은 지시(indicator) 함수이며, w_h는 직립 보행의 가중치이다. $\wedge$는 논리 연산자 'AND'를 의미한다. 따라서, w_f와 w_c는 각각 속도와 행동 크기에 대한 가중치라 할 때, 보상 r_t은 다음과 같이 정의할 수 있다.

$$r_t = w_f \dot{x}_t + h_t - w_c \parallel \boldsymbol{a}_t \parallel_2^2,$$

2-2. 제안 알고리즘

에이전트와 실제 환경의 상호작용을 통해 시간 t까지 생성된 상태들의 집합을 상태 궤적 $\boldsymbol{\tau}_t = (\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_t)$라 하자. 상태 $\boldsymbol{s}_t$의 구성 요소들의 중요도를 반영하기 위한 가중치 행렬을 $\boldsymbol{W}$라 할 때, $\boldsymbol{W}$는 다음과 같이 정의할 수 있으며,

$$\boldsymbol{W} = \mathbf{diag}(w_1, \ldots, w_{17}),$$

diag는 대각 행렬을 의미하고 $w_j > 0\,(j = 1, \ldots, 17)$은 상태의 구성 요소의 가중치를 나타낸다. w_j가 클수록 해당 상태 구성 요소에 높은 가중치가 부여되므로, 에이전트의 학습에 영향 정도를 반영하여 가중치 행렬 $\boldsymbol{W}$를 구성할 수 있다.

상태 궤적 $\boldsymbol{\tau}_t$ 내 모든 상태 $\boldsymbol{s}_t$에 대해 가중치 행렬 $\boldsymbol{W}$이 곱해진 형태의 가중 상태 궤적을 $\tilde{\boldsymbol{\tau}}_t$라 하면, $\tilde{\boldsymbol{\tau}}_t$은 다음과 같이 표현된다.

$$\tilde{\boldsymbol{\tau}}_t = (\boldsymbol{W}\boldsymbol{s}_0, \ldots, \boldsymbol{W}\boldsymbol{s}_t)$$

LLM은 가중 상태 궤적 $\tilde{\boldsymbol{\tau}}_t$을 입력으로 하여, 다음 상태 $\tilde{\boldsymbol{s}}_{t+1}$를 예측한다. 즉, LLM의 상태 전이 예측 함수를 P_{LLM}이라 하면 LLM이 예측한 다음 상태 $\tilde{\boldsymbol{s}}_{t+1}$는 다음과 같이 표현할 수 있다.

$$\tilde{\boldsymbol{s}}_{t+1} = P_{LLM}(s_{t+1} | \tilde{\boldsymbol{\tau}}_t)$$

LLM의 예측을 통해 출력된 $\tilde{\boldsymbol{s}}_{t+1}$는 가중 상태 궤적 $\tilde{\boldsymbol{\tau}}_t$을 기반으로 예측된 값이므로, 실제 환경에서의 행동과 보상을 기반으로 에이전트의 행동 폴리시를 학습하기 위해 가중 행렬 $\boldsymbol{W}$의 역행렬을 취해 예측된 상태를 복원한다. 복원된 상태를 $\hat{\boldsymbol{s}}_{t+1}$라 하면, $\hat{\boldsymbol{s}}_{t+1}$는 다음과 같이 정의된다.

$$\hat{\boldsymbol{s}}_{t+1} = \boldsymbol{W}^{-1} \tilde{\boldsymbol{s}}_{t+1}$$

본 연구에서는 에이전트의 최적의 행동 폴리시 학습에 영향을 미치는 상태 구성 요소인 $\dot{x}_t$ 및 학습 종료에 직결된 상태 구성 요소인 z_t, θ_t에 해당하는 $w_j\,(j = 1, 2, 9)$를 이 외의 가중치 $w_j\,(j \neq 1, 2, 9)$보다 크게 구성된 가중 행렬 $\boldsymbol{W}$을 사용한다.

이를 통해, 제안하는 알고리즘은 에이전트의 보상함수를 직접 알지 못한 상태에서 실제 환경과의 상호작용이 없이 다음 상태를 예측함으로써 에이전트의 행동 폴리시 학습에 필요한 샘플을 생성할 수 있다.

2-3. 실험 세팅 및 결과

본 실험에서는 이족보행 로봇 시뮬레이터인 Walker2d-v4 환경을 고려하였다. 에이전트의 최적 행동 폴리시 학습 알고리즘으로 SAC (Soft Actor-Critic)를 사용하였으며, 실제 환경 동역학을 예측하기 위한 LLM으로는 Llama-3.2-1B을 사용하였다. 상태 궤적에 적용된 가중 행렬 $\boldsymbol{W}$의 가중치를 다음과 같이 설정하였다.

$$w_j = \begin{cases} 1.2 & \text{if } j = 1, 2, 9 \\ 1 & otherwise \end{cases}$$

모든 실험은 5개의 다른 랜덤 시드(seed)로 1,000,000 타임 스텝 동안 진행하였다. 실험에서는 실제 환경 샘플만 사용하는 기존 방법과 제안하는 실제 환경 샘플 및 LLM을 통한 예측 환경 샘플을 사용하는 방법에서 SAC를 통한 학습 성능을 비교하였으며, 기존 방법은 총 256개 샘플을, 제안하는 방법은 256개의 샘플 중 5%는 LLM을 통한 예측 샘플을 사용하였다.

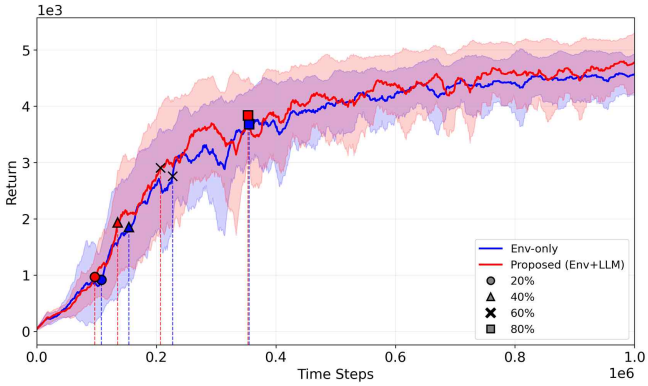

그림 1. 학습에 필요한 샘플의 출처에 따른 에이전트 학습 곡선

그림 1은 기존 방법과 제안 방법을 활용하였을 때, 에이전트의 학습 곡선(learning curve)을 나타낸다. 제안 방법의 평균 성능과 기존 방법의 평균 성능이 유사하게 수렴하는 것으로부터, 제안 방법을 통해 적은 실제 환경 샘플로도 에이전트가 최적의 행동 폴리시를 안정적으로 학습할 수 있는 것을 확인할 수 있다. 또한, 제안된 방법이 초기 학습 단계에서 더 빠른 성능 향상을 보이는 것을 통해 LLM이 에이전트의 보상함수를 직접적으로 알지 않아도 가중 상태 궤적으로 행동 폴리시 학습에 영향을 미치는 환경 동역학을 예측하는 것을 확인할 수 있다.

Ⅲ. 결론

본 논문은 LLM을 활용하여 오프 폴리시 기반 DRL 알고리즘 학습에 필요한 환경 동역학 궤적을 예측하는 방법을 제안하였다. 제안 방법은 행동 폴리시 학습에 상태가 미치는 영향력을 반영하는 가중 행렬을 통해, LLM이 에이전트의 보상 함수를 알지 못하더라도 최적 행동 폴리시 학습에 유리하도록 실제 환경의 동역학을 예측할 수 있다. Walker2d-v4 환경에서의 실험 결과, 제안 방법은 기존의 실제 환경 샘플만을 사용하는 방법과 유사한 평균 성능을 달성함과 동시에 초기 학습 단계에서 더 빠른 성능 향상을 보였다. 이를 통해 제안한 방법이 실제 환경의 샘플 효율성을 확보하면서도 안정적으로 최적 행동 폴리시를 학습할 수 있음을 확인하였다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의지원(No.RS-2021-II210739)과한국연구재단의지원(No.RS-2025-16066913)을 받아 수행된 연구임.

참 고 문 헌

[1] B. Frauenknecht, T. Ehlgen and S. Trimpe, "Data-efficient Deep Reinforcement Learning for Vehicle Trajectory Control," in *Proc. 26th IEEE Int. Conf. Intell. Transp. Syst.*, 2023, pp. 894-901.

[2] A. Benechehab et al., "Zero-shot Model-based Reinforcement Learning using Large Language Models," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025.

[3] H. Tang, D. Key, and K. Ellis, "WorldCoder, a model-based LLM agent: building world models by writing code and interacting with the environment," in *Proc. 38th Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 70148-70212.