УДК 004.89

Л. Р. Кулик[1]
О. Б. Мокін[1]

ПРОБЛЕМАТИКА КОНСИСТЕНТНОСТІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ПРИ НАВЧАННІ З ПІДКРІПЛЕННЯМ

[1] Вінницький національний технічний університет

Анотація

Навчання з підкріпленням (RL), зокрема методи RLHF та DPO, стало стандартом для створення агентів на основі великих мовних моделей (LLM). Однак, ці підходи стикаються з двома фундаментальними проблемами: неконсистентністю поведінки агента та значною обчислювальною неефективністю. Як альтернативне рішення до сучасних підходів запропоновано навчання на основі багатовимірної винагороди та диференційованої критики. Такий підхід дозволяє виконувати цілеспрямоване оновлення політики, системно виправляючи конкретні аспекти поведінки. Це не лише підвищує консистентність агента, але й радикально покращує обчислювальну ефективність, оскільки кожна ітерація навчання несе значно більше інформації. Таким чином, відкриваючи шлях до створення більш надійних, логічно послідовних та ефективних у навчанні автономних агентів.

Ключові слова: глибоке навчання, навчання з підкріпленням, великі мовні моделі, консистентність.

Abstract

Reinforcement learning (RL), particularly methods like RLHF and DPO, has become the standard for creating agents based on large language models (LLMs). However, these approaches face two fundamental problems: inconsistency in agent behavior and significant computational inefficiency. As an alternative to current approaches, training based on multidimensional rewards and differentiated critique is proposed. This approach enables targeted policy updates, systematically correcting specific aspects of behavior. This not only enhances agent consistency but also radically improves computational efficiency, as each training iteration is significantly more information-dense. Thus, this opens the path toward creating more reliable, logically consistent, and training-efficient autonomous agents.

Keywords: deep learning, reinforcement learning, large language models, consistency.

Використання великих мовних моделей (LLM) як основи для автономних агентів є одним з найактивніших напрямів досліджень в галузі штучного інтелекту. Домінантною парадигмою для узгодження поведінки цих моделей з людськими очікуваннями стало навчання з підкріпленням (Reinforcement Learning, RL). Цей процес, як правило, реалізується через дві основні методології: класичне навчання з підкріпленням на основі зворотного зв'язку від людини (Reinforcement Learning from Human Feedback, RLHF) та його більш сучасний і стабільний наступник, пряма оптимізація преференцій (Direct Preference Optimization, DPO) [1]. Ці підходи дозволили створити таких агентів, як Voyager для гри в Minecraft [2] або системи для навігації в інтернеті в середовищі WebArena [3]. Однак, за цією функціональністю ховається фундаментальна проблема: неконсистентність великих мовних моделей (Large Language Models, LLMs), які по своїй природі, є стохастичними генераторами послідовностей, а не детермінованими логічними машинами.

Проблема неконсистентності є багатогранною і проявляється по-різному залежно від ролі LLM. Коли LLM виступає як агент RL, його навчання зазвичай зводиться до оптимізації політики (самої LLM) для максимізації скалярної винагороди. Цей сигнал винагороди надходить від окремої моделі винагороди (Reward Model, RM), навченої на людських преференціях (у випадку RLHF), або безпосередньо виводиться з пар преференцій (у випадку DPO). Однак цей скалярний сигнал є надто обмеженим за своєю інформаційною ємністю і не може охопити всю складність реального світу. Навіть якщо згенерована дія є локально оптимальною і отримує високу винагороду, вона може бути глобально неконсистентною. Це призводить до каскадних помилок у багатоетапному плануванні: невелика галюцинація на ранньому етапі (наприклад, агент вважає, що у нього є необхідний предмет) призводить до невідповідності довгострокового плану. Оскільки модель не має стабільного "стану світу", помилка не коригується, а навпаки, вбудовується в подальші міркування. Сучасні фреймворки, такі як ReAct [4] (Reasoning and Acting) або Reflexion [5], намагаються структурувати

процес мислення агента, але вони є надбудовами над фундаментально неконсистентною моделлю і не вирішують цю базову проблему.

Сьогоднішня практика здебільшого зосереджена на пом'якшенні, а не на вирішенні проблеми неконсистентності. Канонічний підхід, RLHF, як його застосовували в InstructGPT [6] та GPT-4 [7], включає складний багатоетапний процес. Спочатку модель винагороди (Reward model, RM) навчається передбачати, яку з двох відповідей обере людина. Потім, використовуючи алгоритми, такі як Proximal Policy Optimization (PPO) [8], основна LLM, яка виступає політикою (стратегією для визначення наступної дії), доналаштовується для генерації відповідей, що максимізують оцінку від RM. Цей процес є продуктивним, але обчислювально дорогим, нестабільним і страждає від проблеми Out-of-Distribution (OOD), коли політика починає генерувати відповіді, які RM ніколи не бачила і оцінює некоректно. Даний підхід не є в повній мірі ефективним, оскільки сигнал винагороди є слабким та неінформативним, агент змушений покладатися на метод спроб і помилок, що вимагає величезної кількості ітерацій, щоб статистично вивести зв’язок між своїми діями та винагородою [9]. Це вимагає мільйонів ітерацій взаємодії з моделлю та середовищем, що робить процес навчання надзвичайно тривалим та енергозатратним. DPO та його варіації, що використовуються в моделях Llama 3 [10] та Qwen2 [11], переформулюють задачу як просту класифікацію на парах преференцій. Це значно стабілізує навчання. Проте, як і RLHF, DPO все ще оптимізує модель під вузький сигнал людських вподобань, а не під глибоку логічну послідовність.

Всі ці підходи є реактивними, тобто вони намагаються виправити або відфільтрувати неконсистентність після її виникнення, що є обчислювально затратно. Замість намагання виправити неконсистентність через слабкий сигнал, ми пропонуємо змінити саму природу зворотного зв'язку. Наша пропозиція – відмовитись від скалярної винагороди на користь багатовимірного, структурованого сигналу, який ми називаємо диференційованою критикою.

Замість однієї моделі винагороди, що видає число, пропонується використовувати модель-критика (це може бути більше велика LLM, наприклад, GPT-4, або спеціалізована модель). Після того, як основний агент генерує відповідь, критик надає зворотний зв'язок не у вигляді числа, а у вигляді структурованого об'єкта або текстового пояснення, що оцінює відповідь за кількома ключовими характеристиками, такі як: консистентівсть, правдивість, послідовність і т.д.

Найважливіше те, що цей структурований сигнал можна використовувати для прямого обчислення градієнтів для оновлення політики основної моделі. Замість того, щоб використовувати скалярну винагороду в PPO, на основі цієї багатовимірної критики формується вектор втрат. Кожен компонент цього вектора буде відповідати за певний аспект поведінки. Наприклад, низький «consistency_score» згенерує градієнти, що будуть цілеспрямовано модифікувати ваги нейронної мережі, відповідальні за логічні операції та міркування. Навчальний цикл зображено на рисунку 1.

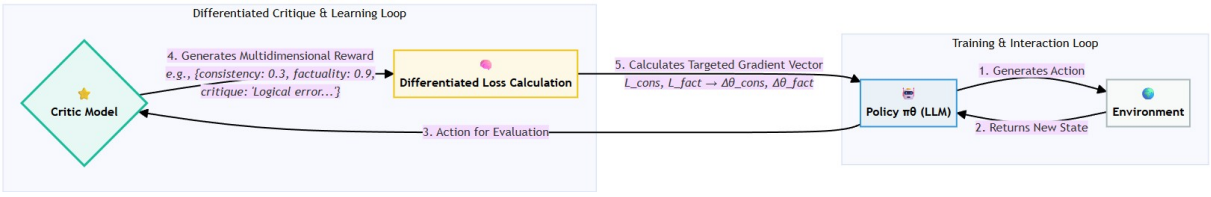

Рисунок 1 – Тренувальний цикл запропонованого підходу

Запропонований підхід має кілька ключових переваг. По-перше, він забезпечує цілеспрямоване навчання, де агент чітко розуміє, який аспект своєї поведінки потрібно виправити, що дозволяє системно покращувати консистентність. По-друге, він радикально підвищує ефективність, оскільки кожна ітерація навчання несе набагато більше інформації, дозволяючи досягти того ж рівня якості за значно меншу кількість кроків. Нарешті, підхід покращує інтерпретованість, оскільки дозволяє відстежувати, як змінюється кожен конкретний аспект поведінки агента.

Висновки

Неконсистентність та обчислювальна неефективність є двома сторонами однієї медалі, що походять від інформаційної бідності скалярної винагороди в сучасних RL-методах. Запропонований

підхід навчання на основі багатовимірної винагороди та диференційованої критики дозволяє одночасно вирішити обидві проблеми, роблячи процес навчання більш цілеспрямованим, швидким та надійним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Shuhe Wang, et al. “Reinforcement Learning Enhanced LLMs: A Survey” arXiv:2412.10400 [cs.CL], Dec. 2024.
2. Guanzhi Wang, et al. “Voyager: An Open-Ended Embodied Agent with Large Language Models” arXiv:2305.16291 [cs.AI], May 2023.
3. Will Maddern, et al. “WebArena: A Realistic Web Environment for Building Autonomous Agents” arXiv:2307.13854 [cs.AI], Jul. 2023.
4. Shunyu Yao, et al. “ReAct: Synergizing Reasoning and Acting in Language Models” arXiv:2210.03629 [cs.CL], Oct. 2022.
5. Noah Shinn, et al. “Reflexion: Language Agents with Verbal Reinforcement Learning” arXiv:2303.11366 [cs.AI], Mar. 2023.
6. Long Ouyang, et al. “Training language models to follow instructions with human feedback” arXiv:2203.02155 [cs.CL], Mar. 2022.
7. OpenAI, et al. “GPT-4 Technical Report” arXiv:2303.08774 [cs.CL], Mar. 2023.
8. John Schulman, et al. “Proximal Policy Optimization Algorithms” arXiv:1707.06347 [cs.LG], Jul. 2017.
9. Wenyun Li, et al. “Shaping Sparse Rewards in Reinforcement Learning: A Semi-supervised Approach” arXiv:2501.19128 [cs.LG], Jan. 2025.
10. Aaron Grattafiori, et al. “The Llama 3 Herd of Models” arXiv:2407.21783 [cs.AI], Jul. 2023.
11. An Yang, et al. “Qwen2 Technical Report” arXiv:2407.10671 [cs.CL], Jul. 2024.

Кулик Леонід Русланович – аспірант факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: leonidkulik2707@gmail.com

Мокін Олександр Борисович – доктор технічних наук, професор, професор кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, Вінниця, e-mail: abmokin@gmail.com

Kulyk Leonid – a graduate student, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, email : leonidkulik2707@gmail.com

Mokin Oleksandr – Dr. Sc. (Eng.), Professor of the Department of System Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: abmokin@gmail.com