ВИБІР АЛГОРИТМУ ДЛЯ НАВЧАННЯ ДРОНА В СЕРЕДОВИЩІ З ДИНАМІЧНИМИ ЗМІНАМИ

Горобець Олександр
здобувач вищої освіти магістерського рівня
Кафедра обчислювальної техніки
НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»
Київ, Україна
Горобець Сергій
кандидат педагогічних наук, доцент
Кафедра комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка
Житомир, Україна

У сучасному світі безпілотні апарати, відомі також як дрони, стали невід'ємною частиною багатьох галузей людської діяльності. Завдяки стрімкому розвитку технологій робототехніки, інформаційно-комунікаційних систем, авіаційної та космічної інженерії, дрони, зокрема, безпілотні літальні апарати (БПЛА) отримали широке застосування. Їх використовують у військовій діяльності, для моніторингу та захисту цивільної інфраструктури, у логістиці для доставки товарів, у сільському господарстві для моніторингу полів та обробки культур, у кінематографі та телебаченні для створення вражаючих кадрів, у наукових дослідженнях та багатьох інших сферах.

Водночас із розширенням можливостей дронів постала потреба в їх автономному управлінні. Це обумовлено тим, що у складних та динамічних середовищах дрони повинні приймати рішення швидше та ефективніше, ніж це можливо під контролем оператора. У цьому контексті розвиток штучного інтелекту (ШІ) відкрив нові перспективи для автоматизації дронів, що зумовлює актуальність вибору методів і алгоритмів штучного інтелекту для тренування дрона в 3D-середовищі з динамічними змінами.

Для створення автономного дрона необхідно обрати найбільш підходящий метод навчання. Серед основних методів виділяють:

1) навчання з учителем (контрольоване навчання), яке базується на попередньо промаркованих даних [1];

2) навчання без учителя (неконтрольоване навчання), орієнтоване на

виявлення закономірностей у даних без використання цільової функції [2];

3) навчання з підкріпленням (Reinforcement Learning, RL), що спрямоване на оптимізацію стратегії агента через взаємодію з середовищем і отримання винагороди за успішні дії [3].

Для визначення оптимального підходу важливо врахувати особливості управління дроном у такому середовищі, а саме:

– стратегічність дій: головна мета дрона – досягти визначеного пункту призначення, а оптимальний шлях досягнення може бути невідомим заздалегідь.

– динамічні зміни середовища, оскільки воно постійно змінюється, наприклад, змінюються погодні умови, виникають нові перешкоди або з'являються рухомі об'єкти.

– складність середовища: дрони діють у складних середовищах, що можуть включати різноманітні перешкоди, наприклад, дерева чи будівлі;

– безперервність простору дій: дрон має виконувати точні маневри, які вимагають безперервних значень дій, наприклад, у вигляді зміни швидкості чи кута нахилу;

Врахування першої особливості означає, що навчання дрона повинне бути орієнтоване на досягнення мети і винагороди, а також потребу в прийнятті послідовних рішень. При цьому контрольоване навчання не оптимізоване на винагороди, оскільки воно просто зіставляє вхідні дані із заздалегідь визначеними результатами, без урахування довгострокових наслідків чи оптимізації з часом.

Неконтрольоване навчання також не передбачає винагороди або цілей. Воно більше зосереджене на пошуку закономірностей у даних, а не на оптимізації цільової функції.

Орієнтацію на досягнення мети та довгострокові наслідки можна отримати лише при навчанні з підкріпленням. Це через те, що воно стимулює приймати оптимальні рішення не лише в конкретний момент часу, але й протягом серії кроків, оскільки агент RL, ігноруючи локальні перспективи, намагається максимізувати сукупну винагороду з часом, яка безпосередньо відповідає досягненню конкретних цілей. Отже, навчання з підкріпленням найкраще підходить для завдань безперервного контролю, таких як навігація безпілотників.

Проблема динамічного і мінливого середовища полягає в тому, що безпілотник повинен постійно адаптуватися до нових перешкод, зміни погодних умов, вітру або появи рухомих цілей. Непередбачуваність такого середовища означає, що попереднє програмування всіх можливих сценаріїв або навчання зі статичним набором даних (як у навчанні з учителем) є недоцільним. Також не підходить і навчання без учителя, оскільки воно в принципі не здатне до динамічної адаптації.

На відміну від них, навчання з підкріпленням за своєю суттю є адаптивним. Агент безперервно навчається під час взаємодії з навколишнім середовищем. Коли середовище змінюється, агент RL може оновити свою політику, щоб врахувати нову динаміку, дозволяючи дрону належним чином

реагувати на непередбачувані або нестандартні ситуації.

Одна з ключових переваг навчання з підкріпленням над навчанням з учителем полягає в тому, що воно самостійно досліджує середовище і вишукує найоптимальніші стратегії. Це особливо важливо при роботі зі складними середовищами, які мають дуже великий простір станів-дій. У цьому випадку воно відносно швидко може знайти стратегії кращі, ніж ті, які будуть підготовлені і промарковані для контрольованого навчання. Отже, врахування третьої умови, а саме, складнощі середовища, також потребує зробити вибір на користь навчання з підкріпленням.

Наступною особливістю є те, що дрони працюють у безперервному просторі дій, де дії не є дискретними, а можуть приймати будь-які значення в межах діапазону.

При навчанні з учителем існує вимога мати попередньо промарковані дані, що робить його неефективним та неточним для такого тренування. А навчання без учителя взагалі не зосереджене на прийнятті рішень чи виконанні дій, що унеможливлює його використання в не дискретних просторах.

Натомість розроблені алгоритми навчання з підкріпленням, які спеціально призначені для обробки безперервних просторів дій, що робить їх ідеальними для керування дроном, де необхідні точні дії.

Таким чином, з огляду на наведені вище особливості, які вимагають від навчального алгоритму гнучкості, адаптивності та орієнтації на ціль, навчання з підкріпленням є найбільш придатним методом. Воно стимулює агента приймати оптимальні рішення не лише в конкретний момент, але й у довгостроковій перспективі, максимізуючи сукупну винагороду. Крім того, RL адаптується до змін у середовищі, що важливо для дронів, які стикаються з новими перешкодами, змінними погодними умовами або рухомими цілями.

Інші підходи мають значні обмеження. Контрольоване навчання вимагає попередньо промаркованих даних, що ускладнює його застосування в умовах динамічних змін. Неконтрольоване навчання, своєю чергою, не спрямоване на прийняття послідовних дій і досягнення конкретних цілей.

Навчання з підкріпленням дозволяє [3]:

– самостійно досліджувати середовище, знаходячи оптимальні стратегії навіть у складних умовах;

– приймати серію рішень, що сприяють досягненню довгострокових цілей;

– адаптуватися до нових ситуацій у реальному часі;

– працювати в безперервному просторі дій, забезпечуючи високу точність управління.

На даний час розроблено багато видів навчання з підкріпленням та алгоритмів, які їх реалізують. Кожен з них підходить для різних типів середовищ і завдань та має свої переваги й обмеження.

Так, безмодельні методи зазвичай використовуються, коли середовище є складним і невідомим, тоді як методи на основі моделей є ефективними, коли взаємодія з середовищем є дорогою. Ціннісні методи зосереджені на оцінці винагороди, тоді як методи, що базуються на політиці, безпосередньо оптимізують рішення агента. Акторно-критичні методи поєднують обидва

підходи, використовуючи їхні сильні сторони.

Порівняльна характеристика найбільш популярних алгоритмів навчання з підкріпленням наведена в табл.1 [4].

Таблиця 1. Ключові характеристики найбільш популярних алгоритмів навчання з підкріпленням

Алгоритм	Тип політики On/ Off; детермінована (Д) /стохастична (С)		Простір станів дискретний (Д)/ неперервний (Н)	Простір дій дискретний (Д)/ неперервний (Н)	Метод розвідки	Стабільність	Ефективність на вибірці	Метод оптимізації
DQN	Off	Д	Д або Н	Д	є-Greedy	нормальна, за певних умов	помірна	ціннісно-орієнтований
DDQN	Off	Д	Д або Н	Д	є-Greedy	більша, ніж у DQN	помірна	ціннісно-орієнтований
SAC	Off	С	Д або Н	Н	Entropy Regularization	висока	висока	Актор-Критик
A2C	On	С	Д або Н	Д або Н	Advantage Function	нормальна	помірна	Актор-Критик
TRPO	On	С	Д або Н	Д або Н	KL-Divergence constraint	нормальна; потребує значних обчислювальних ресурсів	низька	політико-орієнтований
PPO	On	С	Д або Н	Д або Н	Clipped Surrogate Objective	висока	висока	політико-орієнтований
DDPG	Off	Д	Д або Н	Н	Ornstein-Uhlenbeck Process	менша, ніж у TD3	низька	Актор-Критик

Аналіз таблиці дозволяє зробити висновок, що для задачі управління дроном в 3D середовищі з динамічними змінами найкраще підходить алгоритм Proximal Policy Optimization (PPO), розроблений компанією OpenAI у 2020 р.

Алгоритм PPO поєднує простоту реалізації з високою ефективністю навчання. Ключова ідея алгоритму полягає в тому, що він використовує єдину політику і виконує кілька епох оновлень для однієї і тієї ж партії зібраних траєкторій. Відношення ймовірностей обчислюється між поточною політикою (яка оновлюється) і політикою, яка використовувалася для збору траєкторій (до оновлення). Це дозволяє PPO робити консервативні оновлення політики, одночасно покращуючи її продуктивність [5].

Існує дві варіації алгоритму PPO: PPO-penalty та PPO-clip. У першому

використовується поняття розходження Кульбака-Лейблера [6] для оцінки відмінності між новою та старою політикою. PPO вводить обмеження для оновлення політики, щоб запобігти надто великим оновленням, які можуть дестабілізувати процес навчання. Отже, чим більшим є відхилення нової політики від старої, тим більший нараховується штраф. Тим самим алгоритм застерігає агента від внесення великих змін в політику.

Другий варіант PPO, найбільш популярний, замість нараховування штрафів, використовує відсікання у цільовій функції, яке в принципі унеможливлює одномоментну значну зміну політики [5].

Найголовнішою перевагою PPO над іншими алгоритмами навчання з підкріпленням є його стабільність та надійність. PPO запобігає різким змінам у політиці агента, що забезпечує збіжність алгоритму та стійке навчання.

Не менш важливою рисою, особливо для розробників-початківців, є простота налаштування. Порівняно з іншими алгоритмами, PPO, як правило, не вимагає настільки ретельно налаштованих гіперпараметрів. Його стабільність дозволяє йому добре працювати з розумними налаштуваннями за замовчуванням.

Ще однією позитивною рисою PPO є його універсальність та гнучкість, оскільки він може працювати як в дискретному, так і в неперервному просторі дій. Саме тому алгоритм добре підходить для широкого спектра задач, від ігор до управління реальними роботами.

Не можна ігнорувати і його відносну простоту в реалізації. В порівнянні зі своїм попередником – TRPO, він не вимагає рішення таких складних проблем оптимізації, як, наприклад, обмежена оптимізація. Механізм відсікання простий і добре працює в різних задачах.

Незважаючи на ці переваги, PPO має також і недоліки. Зокрема, механізм відсікання, який забезпечує високу стабільність алгоритму, своєю протидією різким змінам політики заважає агенту повністю використати оптимальний напрямок для покращення. Це призводить до більш повільної збіжності або неоптимальних політик порівняно з необмеженими методами. Тому слід зазначити, що оберненою стороною стабільності та обережності в оновленні політики є схильність до потрапляння в локальні оптимуми.

Хоча PPO є більш ефективним на вибірці, ніж класичний VPG, він все ще вимагає великої кількості взаємодій з навколишнім середовищем, що робить його непридатним для задач, де дані є дуже дорогими або де потрібна продуктивність у реальному часі.

Оскільки PPO використовує архітектуру актор-критик, оцінка функції цінності (критика) може стати складною у середовищах з високою дисперсією винагороди або там, де важко вивчити точну функцію цінності. Якщо критик погано навчений, це може негативно вплинути на оновлення політики.

Хоча PPO менш складний, ніж TRPO, він все одно вимагає значних обчислювальних ресурсів, особливо при запуску декількох епох оновлень. Навчання може бути повільним у дуже багатовимірних просторах дій або в середовищах зі складною динамікою.

Таким чином, навчання з підкріпленням є оптимальним методом для розв'язання задачі автономного управління дроном у 3D-середовищі з динамічними змінами. Алгоритм PPO, завдяки своїй стабільності, простоті та універсальності, є найкращим вибором для реалізації цієї задачі. Механізм відсікання забезпечує контрольоване оновлення, що призводить до стійкого навчання, і алгоритм є достатньо універсальним, щоб обробляти як безперервні, так і дискретні дії. Його застосування дозволяє створити ефективну систему, яка здатна адаптуватися до змін, приймати точні рішення у безперервному просторі дій і досягати поставлених цілей навіть у складних умовах.

Список використаних джерел

1. Jiang T., Gradus J. L., Rosellini A. J. Supervised machine learning: A brief primer. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7431677/ (дата звернення: 09.12.2024).
2. What is unsupervised learning? URL: https://www.ibm.com/topics/unsupervised-learning (дата звернення: 09.12.2024).
3. Murel J. What is reinforcement learning? URL: https://www.ibm.com/topics/reinforcement-learning (дата звернення: 09.12.2024).
4. Reinforcement Learning Algorithms: An Overview and Classification. URL: https://arxiv.org/pdf/2209.14940 (дата звернення: 09.12.2024).
5. Osmulski R. Introduction to Proximal Policy Optimization (PPO). URL: https://radekosmulski.com/introduction-to-proximal-policy-optimization-ppo/ (дата звернення: 09.12.2024).
6. Kullback S., Leibler R. A. On Information and Sufficiency. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full (дата звернення: 09.12.2024).