В. А. Челпанов[1], А. В. Бузмаков[2]

[1] Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, г. Санкт-Петербург

[2]Пермский государственный национальный исследовательский университет, г. Пермь

УДК 05.13.18

РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММ ДЛЯ СОЗДАНИЯ СОРЕВНОВАТЕЛЬНОЙ СРЕДЫ СРЕДИ АВТОНОМНЫХ АГЕНТОВ

Аннотация. В статье представлено описание комплекса программ для создания соревновательной среды для автономных агентов. Полученная платформа может быть использована для погружения обучающихся в предметную область и получения ими практического опыта применения алгоритмов обучения с подкреплением в игровой форме.

Ключевые слова: обучение с подкреплением, автономные агенты, соревновательная среда, машинное обучение.

Введение. В последние годы обучение с подкреплением стало одним из наиболее перспективных и динамично развивающихся направлений в области машинного обучения. Эти методы основаны на взаимодействии агента с окружающей средой, в ходе которого агент принимает решения и совершает действия, обучаясь на последствии совершенных действий. Ключевая идея обучения с подкреплением — агент оптимизирует свое поведение посредством проб и ошибок, максимизируя вознаграждение [1].

Решая задачу комплексной оптимизации, обучение с подкреплением находит применение в различных областях, начиная от автономного управления транспортными средствами [2] и заканчивая разработкой сложных алгоритмов для управления энергетическими системами [3], что подчеркивает потенциал применения данной технологии для решения разнообразных задач.

Концепция максимизации награды, ключевая для обучения с подкреплением, обеспечивает основу для создания платформы по организации соревнований между автономными агентами. Интересная и динамичная среда, большое количество потенциальных решений и соревновательный дух могут способствовать эффективному обучению специалистов. Подобные решения позволяют участникам применять теоретические знания на практике. Помимо этого появляется возможность наблюдать за влиянием изменений в стратегиях на поведение агентов в соревновательной среде.

Проблема исследования. Обучение с подкреплением, несмотря на возможность применения в различных областях, сталкивается с рядом ограничений, которые сдерживают его потенциал. Одной из ключевых проблем является сложность перехода от теоретических знаний к практическому применению. Существует потребность в инструментах, позволяющих осуществлять обучение и развитие навыков в практической, а не только теоретической среде. Это подчеркивает необходимость создания платформ, которые могли бы помочь в обучении и развитии способностей людей, работающих с алгоритмами искусственного интеллекта, в частности с обучением с подкреплением.

Это исследование направлено на реализацию платформы для проведения соревнований автономных агентов. Основная цель — создать среду, которая не только улучшит понимание и практическое применение алгоритмов обучения с подкреплением, но и позволит участникам активно разрабатывать, тестировать и совершенствовать своих агентов в контексте соревнования. Задачи исследования включают в себя:

1. Разработку архитектуры платформы: определение технических и функциональных требований к платформе, разработка ее архитектуры и интерфейса пользователя.

2. Реализацию механизмов соревнования: внедрение алгоритмов для проведения соревнований между агентами, включая правила оценки их действий и стратегий.

3. Тестирование и оптимизацию платформы: проведение тестов с участием реальных пользователей для выявления недочетов и оптимизации работы платформы.

На данный момент существует ряд платформ, в которых реализована соревновательная составляющая и возможность применения обучения с подкреплением. Так, например, "*AIcrowd*" [4] на протяжении нескольких лет организует соревнования в рамках крупнейшей конференции *NeurIPS*. Среди отечественных аналогов можно выделить "*all cups*" [5]. Ключевая особенность заключается в том, что соревнование проходит в фиксированный промежуток времени, а среда и правила задаются организаторами. Помимо этого, за счет того, что зачастую все желающие могут участвовать в подобных мероприятиях, уровень конкуренции может быть очень высоким, что может негативно сказаться на процессе погружения в область. Описанные выше факторы не позволяют использовать подобные площадки для решения поставленной задачи в рамках этой работы.

Наиболее близка к поставленным требованиям платформа с открытым исходным кодом *Bot-Games.Fun* [6], на которой участникам предоставляется возможность реализовать своих агентов для участия в соревнованиях. К ее особенностям можно отнести:

– возможность интегрировать свою среду за счет открытого исходного кода и заинтересованности автора проекта;

– для участия в соревновании агента нужно запустить на своей машине и реализовать определенный *API*, с помощью которого он будет взаимодействовать с игровым сервером.

Несмотря на наличие готовой инфраструктуры и способов для интегрирования своих сред, запуск агентов на локальной машине может негативно повлиять на соревновательную составляющую за счет неравномерного распределения технических мощностей у участников.

Таким образом создание инфраструктуры для проведения соревнований среди автономных агентов с возможностью интеграции своих сред, визуализации результатов, развертывания игрового сервера на имеющихся вычислительных мощностях — актуальная на данный момент задача.

Материалы и методы. При реализации всех компонентов использовалось программное обеспечение *Docker*, что позволило сделать систему более модульной, удобной для масштабирования и развертывания. *Docker* обеспечивает изоляцию приложений, благодаря чему каждый компонент системы может работать в собственной среде. Это значительно упрощает управление зависимостями и конфигурациями, а также повышает безопасность, поскольку изменения в одном сервисе не влияют на работу других.

На рис. 1 представлены основные блоки решения. В реализованной системе данные проходят по следующему пути:

– Файл с решением / Модель: пользователь создает файл с решением и модель, которая описывает стратегию участника.

– Телеграм бот для отправки решения: пользователь отправляет файл с решением и модель через телеграм бота.

– Первичная проверка корректности: решение проходит проверку на корректность. Если проверка проходит успешно, данные отправляются в базу решений. В противном случае пользователю приходит подробная информация об ошибке.

– База решений: корректное решение сохраняется в базе данных решений.

– Игровой сервер: решения участников передаются на игровой сервер, где агенты, отправленные участниками, запускаются в среде, для которой они были разработаны.

– Турнирная система: игровой сервер взаимодействует с турнирной системой, где организуются турниры и матчи между различными решениями.

– База турниров: результаты турниров сохраняются в базе данных.

– Визуализация: данные из базы турниров используются для визуализации результатов и персональной статистики.

Таким образом, система обеспечивает полный цикл обработки решений, начиная от их создания и отправки, до участия в турнирах и визуализации результатов. Далее будет представлено подробное описание каждого компонента.

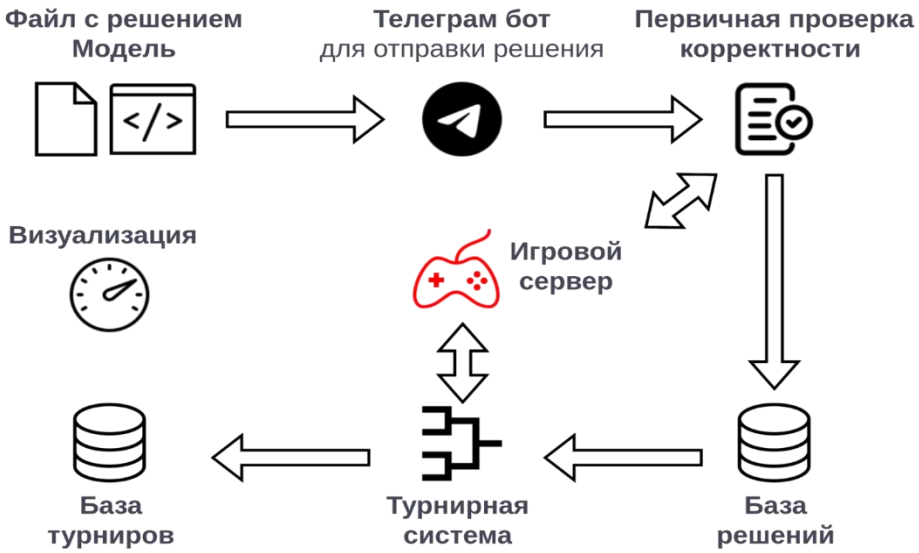

Рис. 1. Общая схема реализованной инфраструктуры соревновательной среды

Взаимодействие участника соревнований, далее пользователя, и платформы организовано с помощью *Telegram* бота. Выбор данного способа взаимодействия с площадкой обусловлен тем, что использование бота позволяет сосредоточить усилия пользователя на логике работы агента и его функциональности, минимизируя затраты и ускоряя процесс разработки. Кроме того, *Telegram* распространен, прост в использовании и доступен на всех популярных платформах и устройствах, что позволяет пользователям взаимодействовать с системой без необходимости устанавливать дополнительные приложения или программы. К основным функциям, реализованным в данном модуле, можно отнести:

– Регистрацию: позволяет пользователям регистрироваться в системе, указав информацию, которая необходима для идентификации.

– Отправку решений: пользователи могут отправлять свои решения в виде кода на *Python* (файл с расширением *.py*) и моделей (файл с расширением *.pth*). Для унификации

архитектуры решений фиксируется интерфейс, который участники должны реализовать для того, чтобы их модуль корректно отрабатывал в общей среде. После отправки решение передается на игровой сервер для проверки на корректность.

– Получение статистики: пользователи могут запросить ссылку на страницу с личной статистикой, что помогает им отслеживать свой прогресс и сравнивать его с другими участниками.

Игровой сервер реализован в виде *API*, который управляет симуляциями игр и оценкой агентов в контролируемой среде. В качестве веб-фреймворка был выбран *FastAPI*, он позволяет получить преимущества асинхронной обработки и автоматической валидации запросов, что повышает надежность и масштабируемость сервиса. Интерфейс для среды симуляции был во многом вдохновлен библиотекой *PettingZoo*, которая воплощает модель взаимодействия *Agent Environment Cycle* [7]. Эта модель позволяет точно воссоздавать динамические взаимодействия между агентами. В *API* представлены две основные конечные точки, которые используются для коммуникации с игровым сервером:

– Конечная точка симуляции: принимает параметры, такие как имя среды, каталог для вывода и детали агентов, затем выполняет симуляцию игры. Использует многопроцессорную настройку для выполнения симуляции в отдельном процессе, захватывая любые исключения и результаты. Результат, включая метрики игры и пути к любым созданным артефактам, возвращается в структурированном ответе, определенном моделями Pydantic.

– Конечная точка тестирования: аналогично симуляции, предназначена для тестирования конкретных агентов в данной среде. Также обрабатывает процесс асинхронно и предоставляет подробную отчетность об ошибках клиенту, если тест не удается из-за ошибок выполнения или ограничений времени.

Следующий компонент системы разработан для проведения турниров среди агентов в игровой среде. С помощью подключения к базе данных, он извлекает необходимую информацию о стратегиях и рейтингах участников. Далее проводятся игры в формате «каждый с каждым». Все игры управляются и отслеживаются с помощью игрового сервера, а результаты сохраняются для последующего анализа и использования. Коммуникация с игровым сервером осуществляется при помощи *API*, который позволяет отправлять запросы на проведение игр, получать результаты и обрабатывать их в реальном времени. После завершения турнира вся информация, включая детали каждой игры, сохраняется либо в базу данных, либо в объектное хранилище. Сбор информации позволяет не только вести детальный учет прошедших соревнований, но и анализировать эффективность агентов в различных условиях, способствуя улучшению стратегий и подходов в будущих итерациях. Так, например, на базе сыгранных турнирных игр в режиме один на один рассчитывается рейтинг *ELO* [8], который широко распространен в шахматах. В зависимости от формата поведения (личный, командный, и т.д.), могут использоваться различные подходы для расчета рейтинга.

Заключительным компонентом системы является интерактивный веб-интерфейс с визуализацией, который позволяет пользователям просматривать и анализировать данные о прошедших турнирах. Он получает и агрегирует информацию, представленную в базе данных и объектном хранилище. Пользователь получает доступ к результатам проведенных турниров, записям игр с участием агента, информации об отправленных стратегиях, графику изменения рейтинга.

На рис. 2 представлена информационная панель для анализа результатов. В левой части размещена подробная информация про выбранный турнир: таблица с результатами, записи игр. Используя эту информацию, участник может изучать сильные и слабые стороны решения. Правая часть изображения фокусируется на сводном анализе стратегии участника. В верхней части представлен график, отображающий позицию участника в турнире и общее количество участников в прошедших турнирах. В нижней части содержится информация о загруженных решениях с указанием времени и краткими примечаниями. Данная информация помогает выполнять сравнительный анализ решений и учитывать общую динамику.

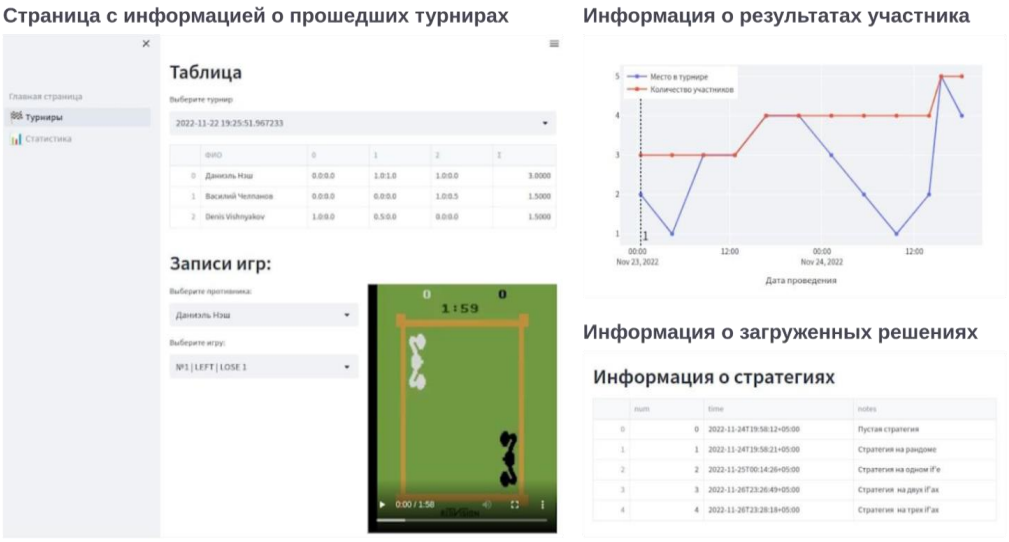

Рис. 2. Демонстрация компонентов веб-интерфейса

Еще один график, содержащий информацию об изменения рейтинга, представлен на рис. 3. Синяя линия представляет точный рейтинг участника в каждый момент времени. Красная пунктирная линия показывает усредненный рейтинг, предоставляя сглаженную версию тренда для лучшего понимания общей динамики. Вертикальные линии на графике соответствуют версии отправленной стратегии, что позволяет оценить, как изменение стратегий влияло на рейтинг участника.

Рис. 3. График изменения рейтинга

Результаты. Тестирование разработанной системы проходило в рамках курса «Введение в искусственный интеллект на *PyTorch*» в ИТ-Университете. В рамках итогового проекта необходимо было разработать своего агента для участия в соревновании по боксу между агентами. Соревнование состояло из последовательности круговых турниров, которые проводили с заданной периодичностью (раз в 4 часа). Всю неделю участники могли дорабатывать своего агента, им предоставлялась возможность отправить неограниченно количество решений в систему, но в турнире участвовало только одно — актуальное на момент начала. Завершалось соревнование расширенным круговым турниром, по итогам которого распределялись финальные места.

Система успешно прошла тестирование, за 7 дней было проведено порядка 50 круговых турниров средняя продолжительность которых составила 2 часа, сыграно порядка 2000 игр, в соревновании приняли участие 14 человек. Была получена положительная обратная связь от участников, которая подтверждает гипотезу о возможности применения соревновательных сред для погружения в область обучения с подкреплением. Эти результаты демонстрируют потенциал системы для использования в образовательных и исследовательских целях.

К преимуществам полученной системы можно отнести:

- Модульность: система построена на модульной архитектуре, что позволяет легко добавлять, удалять или изменять отдельные компоненты без воздействия на работу всей системы. Это облегчает тестирование отдельных функций и интеграцию новых инструментов или алгоритмов, делая платформу гибкой.
- Масштабируемость: благодаря использованию таких технологий, как контейнеризация, платформа легко масштабируется для обработки большего количества пользователей и более сложных симуляций.
- Возможность интеграции различных турнирных сеток: в систему заложен функционал для поддержки различных форматов соревнований, такие как круговая система, олимпийская система. Это позволяет пользователям выбирать наиболее подходящий формат в зависимости от среды и количества участников.

Заключение. Реализованная платформа подтвердила свою востребованность, ее использование позволило участникам активно взаимодействовать, разрабатывать и тестировать свои алгоритмы в динамичной и конкурентной среде. Это не только способствовало глубокому погружению в область обучения с подкреплением, но и позволило участникам на практике освоить новые технологии и подходы.

В рамках дальнейшего развития системы планируется сосредоточить усилия на улучшении производительности и расширении функционала. Это включает в себя оптимизацию существующих процессов, интеграцию новых технологий для более эффективной обработки данных и управления соревнованиями. Кроме того, большое внимание будет уделено написанию подробной документации, что сделает платформу более доступной для новых пользователей и разработчиков, желающих адаптировать систему под свои задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Arulkumaran K. [et al.]. Deep reinforcement learning: A brief survey // IEEE Signal Processing Magazine. — 2017. — Т. 34, № 6. — С. 26-38.
2. Kiran B. R. et al. Deep reinforcement learning for autonomous driving: A survey // IEEE Transactions on Intelligent Transportation Systems. — 2021. — Т. 23, № 6. — С. 4909-4926.
3. Perera A. T. D., Kamalaruban P. Applications of reinforcement learning in energy systems // Renewable and Sustainable Energy Reviews. — 2021. — Т. 137. — С. 110618.
4. AIcrowd. — URL: https://www.aicrowd.com/ (дата обращения: 12 мая 2024 г.).
5. All cups. — URL: https://cups.online/ (дата обращения: 12 мая 2024 г.).
6. Bot-Games.Fun. — URL: https://bot-games.fun (дата обращения: 12 мая 2024 г.).
7. Terry J. et al. Pettingzoo: Gym for multi-agent reinforcement learning // Advances in Neural Information Processing Systems. — 2021. — Т. 34. — С. 15032-15043.
8. Albers P. C. H., de Vries H. Elo-rating as a tool in the sequential estimation of dominance strengths // Animal Behaviour. — 2001. — Т. 61, № 2. — С. 489-495.