УДК 004.8

Кононов Кирилл Дмитриевич

магистр, Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова, г. Новочеркасск

ЭВОЛЮЦИЯ МЕТОДОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ: ПЕРЕХОД ОТ ОДНОАГЕНТНЫХ АЛГОРИТМОВ К МНОГОАГЕНТНЫМ СИСТЕМАМ

Аннотация. В статье рассматривается эволюция методов обучения с подкреплением от классических одноагентных алгоритмов к современным архитектурам глубокого и многоагентного обучения. Проведён сравнительный анализ алгоритмов Q-learning и SARSA, рассмотрены особенности глубокого обучения с подкреплением на примере методов DQN и PETS, а также исследованы современные подходы к организации взаимодействия нескольких агентов, включая архитектуры DRON и централизованное обучение с децентрализованным исполнением (CTDE). Проанализированы основные преимущества, ограничения и области применения различных классов алгоритмов обучения с подкреплением. Представлены сравнительные характеристики рассмотренных подходов, отражающие их эффективность, масштабируемость, вычислительную сложность и способность к адаптации в динамических средах. Показано, что развитие обучения с подкреплением сопровождается переходом от поиска оптимальной стратегии отдельного агента к построению распределённых интеллектуальных систем, ориентированных на коллективное принятие решений и взаимодействие в сложных условиях.

Ключевые слова: обучение с подкреплением, глубокое обучение с подкреплением, многоагентное обучение с подкреплением, Q-learning, SARSA, DQN, PETS, DRON, CTDE, MADDPG.

Annotation. The article examines the evolution of reinforcement learning methods from classical single-agent algorithms to modern deep and multi-agent

learning architectures. A comparative analysis of the Q-learning and SARSA algorithms is presented, followed by a review of deep reinforcement learning approaches based on DQN and PETS, as well as modern multi-agent interaction architectures, including Deep Reinforcement Opponent Network (DRON) and Centralized Training with Decentralized Execution (CTDE). The advantages, limitations, and application domains of different reinforcement learning approaches are analyzed. Comparative characteristics of the considered methods are presented with respect to adaptability, scalability, computational complexity, and learning efficiency in dynamic environments. The study demonstrates that the evolution of reinforcement learning is characterized by a transition from optimizing the behavior of a single agent to developing distributed intelligent systems capable of coordinated decision-making and cooperation in complex multi-agent environments.

Keywords: reinforcement learning, deep reinforcement learning, multi-agent reinforcement learning, Q-learning, SARSA, DQN, PETS, DRON, CTDE, MADDPG.

1. Введение. В последние годы обучение с подкреплением (Reinforcement Learning, RL) стало одним из наиболее активно развивающихся направлений искусственного интеллекта, обеспечивающим создание автономных систем, способных самостоятельно принимать решения на основе взаимодействия с окружающей средой. В отличие от традиционных методов машинного обучения, ориентированных преимущественно на обработку заранее подготовленных данных, обучение с подкреплением позволяет агенту формировать стратегию поведения посредством последовательного получения обратной связи в виде вознаграждений [1]. Такой подход обеспечивает применение интеллектуальных систем в задачах управления, робототехники, автономного транспорта, игровых сред и распределённых вычислительных систем.

Первые эффективные методы обучения с подкреплением были основаны на табличных алгоритмах оценки функции ценности, среди которых особое

место занимают Q-learning и SARSA. Данные методы сформировали теоретическую основу современной области RL и позволили исследовать фундаментальные принципы обучения агентов в условиях неопределённости [2, 3]. Однако их применение ограничено сложностью пространства состояний и действий, что становится критическим фактором при работе с высокоразмерными данными и сложными динамическими средами.

Развитие методов глубокого обучения привело к появлению глубокого обучения с подкреплением (Deep Reinforcement Learning, Deep RL), в котором нейронные сети используются для аппроксимации функций ценности или политики агента. Архитектуры, основанные на Deep Q-Network (DQN), модельно-ориентированных методах, таких как Probabilistic Ensembles with Trajectory Sampling (PETS), и гибридных подходах, расширили область применения RL на задачи, ранее недоступные классическим алгоритмам [4]. Вместе с тем глубокие методы обучения с подкреплением столкнулись с новыми ограничениями, включая высокую вычислительную сложность, низкую эффективность использования данных и проблемы устойчивости обучения.

Следующим этапом развития области стало многоагентное обучение с подкреплением (Multi-Agent Reinforcement Learning, MARL), в котором несколько интеллектуальных агентов одновременно взаимодействуют в общей среде и формируют собственные стратегии поведения. Основные сложности MARL связаны с нестационарностью среды, координацией действий и необходимостью обмена информацией между агентами [5]. В отличие от одноагентных систем, где среда рассматривается как относительно стабильная, в многоагентных сценариях поведение каждого участника влияет на динамику окружающего пространства. Это приводит к появлению дополнительных задач, связанных с координацией действий, коммуникацией между агентами, ограниченностью наблюдений и нестационарностью среды.

Современные архитектуры многоагентного обучения, включая методы централизованного обучения с децентрализованным исполнением (Centralized

Training with Decentralized Execution, CTDE), модели взаимодействия с оппонентами и коммуникационные подходы, направлены на преодоление ограничений классических алгоритмов RL. Однако увеличение количества агентов приводит к росту сложности обучения, расширению пространства совместных действий и необходимости разработки эффективных механизмов обмена информацией.

Развитие обучения с подкреплением можно рассматривать как последовательный переход от одноагентных табличных методов к глубоким архитектурам и далее к распределённым многоагентным системам. Каждый этап расширяет возможности применения RL, но одновременно формирует новые исследовательские задачи, связанные с масштабируемостью, адаптивностью и устойчивостью интеллектуальных систем.

Целью данной работы является сравнительный анализ методов обучения с подкреплением при переходе от одноагентных к многоагентным системам, а также выявление ключевых особенностей, преимуществ и ограничений современных архитектур. Для достижения поставленной цели решаются следующие задачи:

- анализ фундаментальных одноагентных алгоритмов обучения с подкреплением Q-learning и SARSA;
- рассмотрение развития методов глубокого обучения с подкреплением и их роли в расширении области применения RL;
- исследование современных подходов многоагентного обучения, включая архитектуры коммуникации и централизованного обучения;
- проведение сравнительного анализа возможностей и ограничений различных классов RL-методов.

2. Эволюция методов обучения с подкреплением: от классических алгоритмов к глубокому RL. Обучение с подкреплением прошло несколько этапов развития, связанных с увеличением сложности решаемых задач и необходимостью работы с более высокоразмерными пространствами состояний. Первоначально методы RL основывались на табличной оценке

функции ценности и применялись преимущественно в средах с ограниченным количеством состояний и действий [1, 2]. Рост сложности практических приложений, включая робототехнику, автономные системы управления и интеллектуальных агентов, потребовал перехода к методам, способным работать с большими объёмами данных и сложными наблюдениями.

Одним из ключевых направлений развития стало объединение алгоритмов обучения с подкреплением с методами глубокого обучения. Использование нейронных сетей позволило заменить табличное представление функций ценности параметризованными моделями, что значительно расширило область применения RL [4]. В дальнейшем развитие данного направления привело к появлению многоагентного обучения с подкреплением, в котором несколько агентов одновременно взаимодействуют друг с другом и окружающей средой.

В данной работе рассматриваются основные этапы развития RL: от классических алгоритмов Q-learning и SARSA до современных методов глубокого обучения с подкреплением. Особое внимание уделяется анализу различий между подходами, их преимуществам и ограничениям, поскольку именно эти особенности определяют возможность применения алгоритмов при переходе к многоагентным системам.

2.1 Классические алгоритмы Q-learning и SARSA. Классические методы обучения с подкреплением основаны на использовании марковского процесса принятия решений (Markov Decision Process, MDP), включающего множество состояний S, множество возможных действий A, функцию переходов P, функцию вознаграждения R и коэффициент дисконтирования γ. Цель агента заключается в формировании политики поведения π, максимизирующей ожидаемую суммарную награду.

Одними из наиболее распространённых алгоритмов данного класса являются Q-learning и SARSA. Несмотря на общую математическую основу, они используют различные стратегии обновления функции ценности, что

приводит к различиям в поведении агента. Q-learning является модельно-свободным алгоритмом с обучением вне текущей стратегии поведения. Метод оценивает оптимальную функцию ценности действий Q(s, a), независимо от текущей стратегии агента [2]. Обновление выполняется следующим образом:

(1)

где Q(s,a) — текущая оценка ценности выполнения действия a в состоянии s;

α – коэффициент скорости обучения;

r – полученное вознаграждение после выполнения действия;

γ – коэффициент дисконтирования будущих наград;

s′ – новое состояние среды после выполнения действия;

a′ – возможное действие в новом состоянии;

$\max_{a'} Q(s', a')$ – максимальная ожидаемая ценность следующего состояния.

Главным преимуществом Q-learning является способность находить оптимальную стратегию независимо от исследовательского поведения агента. Однако использование максимального значения функции ценности может приводить к переоценке будущих наград, особенно в условиях неопределённости.

Алгоритм SARSA относится к классу методов, в которых обучение выполняется непосредственно на основе используемой агентом стратегии (on-policy learning). В отличие от методов, ориентированных на поиск максимально возможного вознаграждения независимо от текущей политики, SARSA обновляет функцию ценности с использованием действия, которое агент фактически выбирает в следующем состоянии. Благодаря этому процесс обучения тесно связан с реальным поведением агента и учитывает влияние исследовательской стратегии, например ε-жадного выбора действий. Такой механизм позволяет получать более реалистичную оценку ожидаемой награды в условиях неопределённости и повышает устойчивость процесса обучения.

$$Q(s, a) \leftarrow Q(s, a) + \alpha[\, r + \gamma Q(s', a') - Q(s, a) \,] \quad (2)$$

где Q(s,a) — текущая оценка ценности выполнения действия a в состоянии s;

α – коэффициент скорости обучения;

r – полученное вознаграждение после выполнения действия;

γ – коэффициент дисконтирования будущих наград;

Q(s′, a′) – значение ценности следующего состояния и фактически выбранного действия.

В отличие от Q-learning, SARSA учитывает влияние исследовательского поведения агента, поэтому формирует более осторожную стратегию принятия решений. Это делает алгоритм эффективным в задачах, где выполнение потенциально рискованных действий может приводить к значительным потерям или нарушению требований безопасности. Вместе с тем использование фактически выбранных действий может несколько замедлять процесс поиска оптимальной политики по сравнению с Q-learning, однако обеспечивает более стабильную сходимость в динамических и зашумлённых средах. Для количественного сравнения алгоритмов можно использовать показатели эффективности обучения, представленные в таблице 1.

Таблица 1. Сравнение Q-learning и SARSA по основным характеристикам

Table 1. Comparison of Q-learning and SARSA based on evaluation indicators

Показатель оценки (%) **Evaluation indicator**	**Q-learning**	**SARSA**
Скорость обучения Learning speed	92	84
Точность формирования стратегии Strategy formation accuracy	95	88
Устойчивость к шуму среды Robustness to environmental noise	74	91
Эффективность исследования пространства состояний State-space exploration efficiency	89	82
Ошибка оценки функции ценности Value function estimation error	15	8
Требуемое количество взаимодействий со средой Required number of environment interactions	100	115
Устойчивость при изменении политики поведения	78	92

Stability under behavior policy changes		
Склонность к переоценке функции ценности Tendency to overestimate the value function	17	7

Представленные показатели демонстрируют различия между алгоритмами. Q-learning обеспечивает более высокую скорость обучения и эффективность поиска оптимальной стратегии благодаря использованию максимального значения функции ценности следующего состояния. Однако такой механизм увеличивает вероятность переоценки значений Q-функции. SARSA показывает меньшую скорость обучения, но обладает большей устойчивостью к изменениям среды и сниженной ошибкой оценки за счёт использования фактически выбранного действия при обновлении стратегии.

2.2 Переход к глубокому обучению с подкреплением. Развитие глубоких нейронных сетей позволило заменить табличное представление функций ценности приближенными моделями. В Deep RL функция ценности представляется параметризованной нейронной с етью:

$$Q(s, a; \theta) \approx Q(s, a) \quad (3)$$

где: $Q(s, a; \theta)$ – значение функции ценности, вычисленное нейронной сетью;

$Q(s, a)$ – оптимальное значение функции ценности.

θ – параметры нейронной сети;

Одним из наиболее значимых методов стал алгоритм DQN, объединяющий Q-learning и глубокие нейронные сети. Использование целевой сети и буфера воспроизведения опыта позволило повысить стабильность обучения и применять RL к задачам обработки сложных сенсорных данных [4]. Для повышения стабильности обучения используются два основных механизма: целевая сеть и буфер воспроизведения опыта. Ошибка обучения DQN определяется выражением:

$$L(\theta) = \mathbb{E}[(r + \gamma \max_{a'} Q(s', a'; \theta^{-}) - Q(s, a; \theta))^2] \quad (4)$$

где $L(\theta)$ – функция ошибки обучения;

E – математическое ожидание по набору обучающих примеров;

r – полученное вознаграждение;

γ – коэффициент дисконтирования;

θ^- – параметры целевой сети;

θ – параметры основной сети.

Использование DQN позволило применять RL к задачам с изображениями и сложными сенсорными данными. Однако данный подход сохраняет ряд ограничений: необходимость большого количества данных, нестабильность обучения и сложность переноса полученной стратегии в новые условия.

Для решения проблемы эффективности обучения были разработаны модельно-ориентированные методы, одним из наиболее известных представителей которых является алгоритм PETS. В отличие от классических безмодельных методов, PETS использует ансамбль вероятностных моделей динамики среды для прогнозирования будущих состояний и планирования последовательности действий [6]. Архитектура алгоритма включает ансамбль моделей переходов, механизм распространения неопределенности посредством сэмплирования траекторий и модуль планирования на основе управления с прогнозированием модели (Model Predictive Control, MPC), что позволяет существенно сократить количество необходимых взаимодействий со средой. Общая структура алгоритма PETS представлена на рисунке 1.

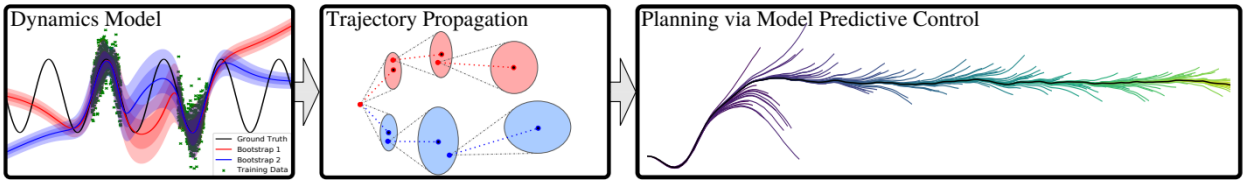

Рис. 1 Архитектура алгоритма PETS, демонстрирующая ансамбль вероятностных моделей и распространение неопределенности

Fig. 1 Architecture of the PETS algorithm illustrating the probabilisticensemble model and uncertainty propagation through trajectory sampling

Дополнительным направлением стали гибридные методы, объединяющие преимущества модельно-ориентированного и безмодельного обучения, например Temporal Difference Models (TDM), использующие целеобусловленные функции ценности для одновременного обучения и планирования.

Таким образом, развитие глубокого обучения с подкреплением позволило перейти от небольших дискретных задач к сложным реальным приложениям. Однако увеличение количества взаимодействующих объектов показало необходимость дальнейшего развития методов RL, что привело к появлению многоагентных архитектур, рассматриваемых далее.

3. Переход к многоагентному обучению с подкреплением. Расширение области применения обучения с подкреплением привело к необходимости решения задач, в которых одновременно взаимодействуют несколько интеллектуальных агентов. К таким задачам относятся автономный транспорт, робототехнические комплексы, распределённые вычислительные системы, многопользовательские игры и коллективное управление беспилотными устройствами. В подобных условиях каждый агент принимает решения не только на основе состояния окружающей среды, но и с учётом поведения остальных участников. Основные особенности MARL связаны с нестационарностью среды, ростом пространства совместных действий и необходимостью координации поведения участников [5].

В отличие от одноагентного обучения, где изменения среды определяются действиями единственного агента, в многоагентном обучении с подкреплением сама среда становится динамической. Изменение стратегии одного агента влияет на условия обучения остальных, что приводит к появлению новых задач координации, обмена информацией и совместного принятия решений.

3.1 Особенности многоагентных RL-систем. Переход от одноагентного к многоагентному обучению сопровождается рядом принципиальных изменений. Если в классическом RL основной задачей является поиск оптимальной стратегии одного агента, то в MARL необходимо учитывать взаимное влияние нескольких одновременно обучающихся участников. Основными особенностями многоагентных систем являются:

- нестационарность среды вследствие непрерывного изменения стратегий других агентов;
- экспоненциальный рост пространства совместных действий;
- необходимость координации поведения при решении задачи;
- частичная наблюдаемость, при которой каждый агент обладает лишь ограниченной информацией о состоянии среды.

Перечисленные особенности существенно усложняют применение классических алгоритмов обучения с подкреплением. В результате возникла необходимость разработки специализированных архитектур, позволяющих учитывать действия других участников и обеспечивать согласованное коллективное поведение. Одним из наиболее известных решений данной проблемы стала архитектура Deep Reinforcement Opponent Network (DRON), предназначенная для моделирования поведения других агентов [7].

3.2 Архитектуры взаимодействия агентов. В отличие от традиционных алгоритмов глубокого обучения с подкреплением, в которых агент принимает решение только на основе собственного состояния, архитектура DRON дополнительно использует информацию о предполагаемом поведении других участников. Такой подход позволяет повысить качество принимаемых решений в условиях совместного обучения и уменьшить влияние нестационарности среды. Архитектура DRON включает две основные модификации, представленные на рисунке 2.

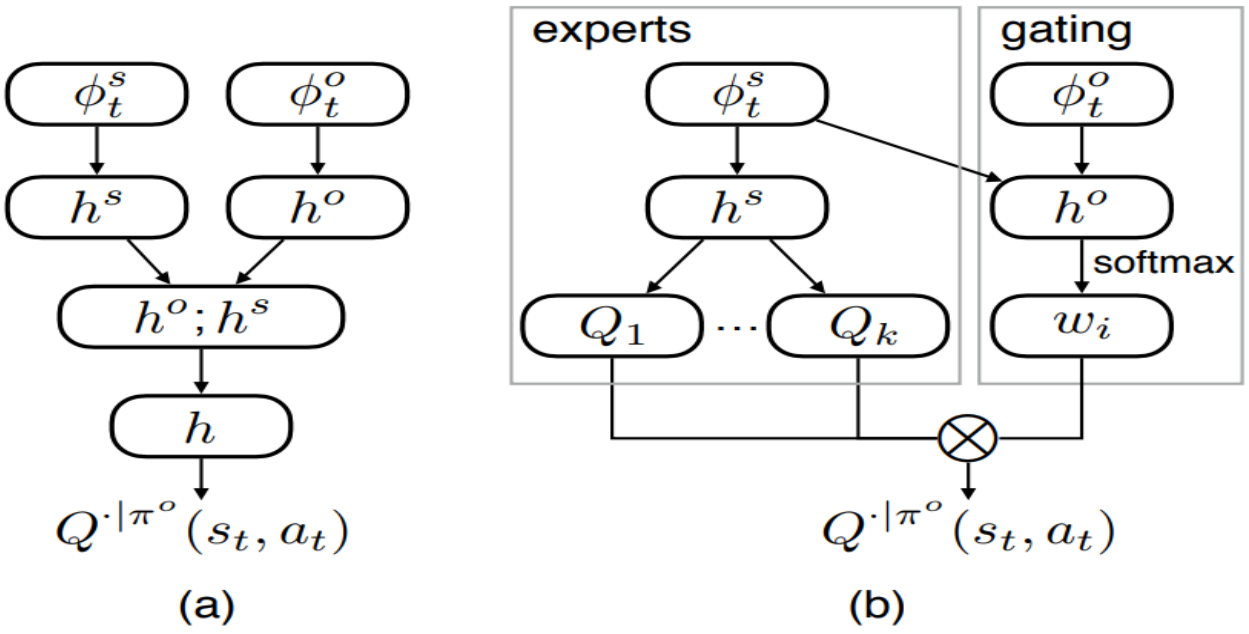

Рис. 2 Архитектура DRON: (a) DRON-concat; (b) DRON-MOE

Fig. 2. DRON architecture: (a) DRON-concat; (b) DRON-MOE

Модель DRON-concat объединяет признаки текущего состояния агента и информацию о предполагаемом поведении оппонента в единый вектор, используемый для вычисления функции ценности. Такой подход отличается простотой реализации и невысокими вычислительными затратами, однако менее эффективен при моделировании сложных стратегий взаимодействия.

Архитектура DRON-MOE (Mixture of Experts) использует несколько специализированных моделей-экспертов, результаты которых объединяются для формирования итогового решения. Это позволяет лучше учитывать различные стратегии поведения других агентов и повышает точность прогнозирования, хотя требует больших вычислительных ресурсов. Сравнение архитектур DRON представлено в таблице 2.

Таблица 2. Сравнение архитектур DRON-concat и DRON-MOE

Table 2. Generation results for various categories of user queries

Показатель оценки Evaluation indicator	**DRON-concat**	**DRON-MOE**
Точность распознавания стратегии оппонента (%)	85	94

Opponent strategy recognition accuracy (%)		
Успешность кооперации агентов (%) Cooperative task success rate (%)	82	91
Скорость адаптации к изменению поведения (%) Adaptation speed to policy changes (%)	84	93
Устойчивость при частичной наблюдаемости (%) Robustness under partial observability (%)	79	90
Среднее число итераций до сходимости Average convergence iterations	2350	1920
Время обработки одного шага (мс) Processing time per decision step (ms)	7.8	11.6
Использование памяти (МБ) Memory usage (MB)	148	224
Общая эффективность взаимодействия (%) Overall interaction efficiency (%)	83	92

Представленные результаты показывают, что архитектура DRON-MOE превосходит DRON-concat по точности распознавания стратегии оппонента, успешности кооперации и скорости адаптации. Использование нескольких экспертов позволяет эффективнее учитывать различные модели поведения агентов, однако приводит к увеличению вычислительной сложности, времени обработки и объёма памяти. В свою очередь, DRON-concat отличается более простой структурой и меньшими требованиями к ресурсам.

Дальнейшее развитие MARL привело к появлению парадигмы централизованного обучения с децентрализованным исполнением. В рамках данного подхода во время обучения используется централизованный критик, учитывающий состояния и действия всех агентов, а после обучения каждый агент действует на основе собственной локальной политики [8]. Это снижает влияние нестационарности среды и повышает масштабируемость систем. Архитектура CTDE на основе алгоритма многоагентного глубокого детерминированного градиента политики (Multi-Agent Deep Deterministic Policy Gradient, MADDPG) представлена на рисунке 3.

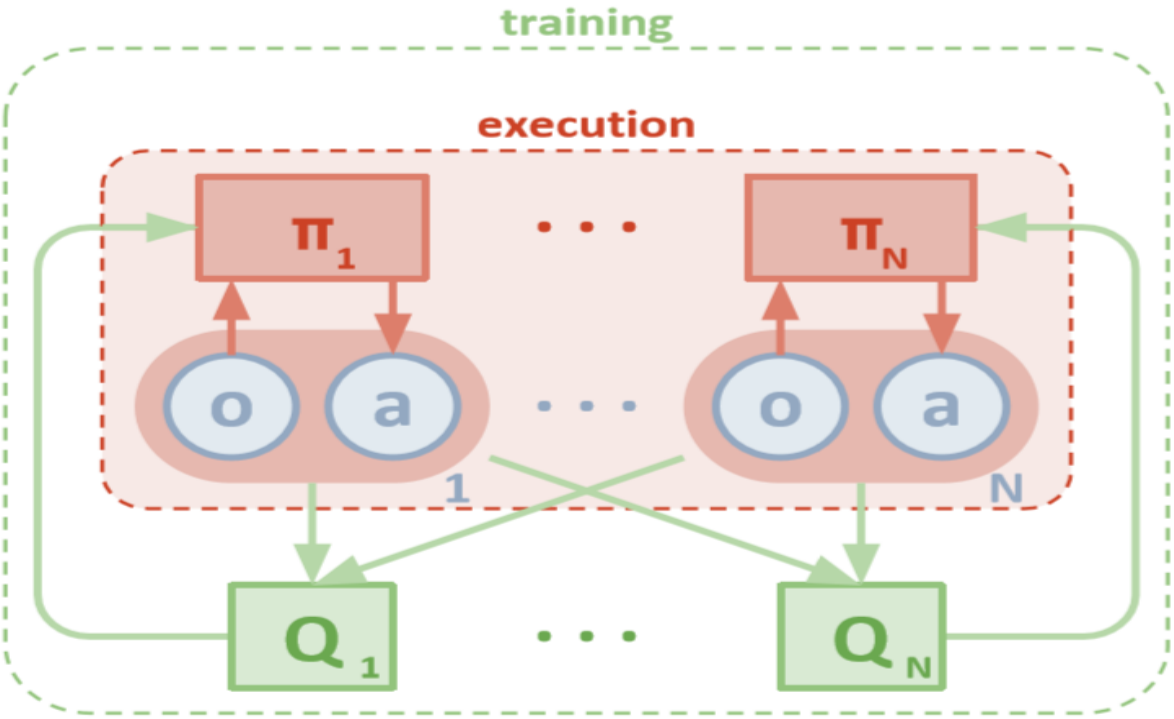

Рис. 3 Архитектура централизованного обучения с децентрализованным исполнением (CTDE) на основе MADDPG

Fig. 3. Centralized Training with Decentralized Execution (CTDE) architecture based on MADDPG

В данной архитектуре центральный критик использует совместную информацию от всех агентов во время обучения, включая состояния и выполняемые действия. При этом после завершения процесса обучения каждый агент функционирует независимо, используя собственную политику, что позволяет применять полученную модель в распределённых системах без постоянного обмена полной информацией.

Использование CTDE стало основой для создания современных алгоритмов MARL, таких как MADDPG, QMIX и COMA, которые объединяют преимущества централизованного анализа данных и децентрализованного принятия решений [8, 9]. Благодаря этому подобные архитектуры применяются в задачах коллективной робототехники, автономного управления транспортом и распределённых интеллектуальных системах.

4. Сравнительный анализ архитектур одноагентного и многоагентного обучения с подкреплением. После рассмотрения этапов развития RL можно выделить несколько классов алгоритмов, различающихся уровнем взаимодействия агентов, способом представления среды и требованиями к вычислительным ресурсам. Классические методы ориентированы на обучение одного агента, тогда как MARL предназначено для задач с несколькими взаимодействующими участниками.

Алгоритмы Q-learning и SARSA формируют основу оценки функции ценности и поиска оптимальной стратегии, однако ограничены при работе со сложными средами и не учитывают поведение других агентов. Методы Deep RL, такие как DQN и PETS, позволяют обрабатывать высокоразмерные состояния, а архитектуры DRON и CTDE обеспечивают адаптацию к поведению других участников и формирование коллективных стратегий. Результаты сравнительного анализа представлены в таблице 3.

Таблица 3. Сравнение архитектур одноагентного и многоагентного RL

Table 3. Comparison of single-agent and multi-agent RL architectures

Показатель оценки Evaluation indicator	**Q-learning**	**SARSA**	**DQN**	**PETS**	**DRON**	**CTDE (MADDPG)**
Тип обучения Learning type	Одноагентный RL Single-agent RL	Одноагентный RL Single-agent RL	Глубокий RL Deep RL	Модельно-ориентированный RL Model-based RL	Многоагентный RL Multi-Agent RL	Многоагентный RL Multi-Agent RL
Адаптивность к изменениям (%) Adaptability (%)	65	72	80	88	90	94
Масштабируемость (%) Scalability (%)	55	58	75	82	78	91
Вычислительная эффективность (%) Computational efficiency (%)	95	92	60	70	65	58
Учет взаимодействия агентов (%) Agent interaction modeling (%)	0	0	0	0	88	95
Эффективность работы с высокоразмерными состояниями (%) High-dimensional state processing efficiency (%)	35	40	92	90	88	93
Требуемый объём данных для обучения (%) Training data requirements (%)	85	82	45	75	55	60
Сложность реализации (%) Implementation complexity (%)	20	25	55	75	80	90

Представленные результаты показывают постепенное усложнение архитектур обучения с подкреплением. Q-learning и SARSA демонстрируют высокую вычислительную эффективность благодаря простой табличной структуре, однако практически не подходят для задач с большим количеством состояний или взаимодействующих объектов. Переход к глубокому RL значительно расширил область применения алгоритмов за счёт использования нейронных сетей, но привёл к росту вычислительных затрат.

Методы PETS показывают преимущества модельно-ориентированного подхода за счёт более эффективного использования данных среды, однако требуют построения дополнительной модели динамики. В многоагентных системах DRON и CTDE обеспечивают возможность учитывать поведение других участников, что является ключевым фактором при решении кооперативных и конкурентных задач [7, 8]. При этом повышение адаптивности достигается за счёт увеличения сложности архитектуры.

Одним из наиболее важных направлений дальнейшего развития MARL является создание масштабируемых архитектур, способных работать с большим количеством агентов, а также совершенствование механизмов коммуникации и обмена информацией [5, 10]. При увеличении количества участников экспоненциально возрастает пространство совместных состояний и действий, что требует разработки более эффективных методов представления информации и распределения вычислений.

Перспективным направлением является развитие механизмов коммуникации между агентами, позволяющих автоматически формировать эффективные способы обмена информацией и координации действий. Значительный потенциал также имеют гибридные методы, объединяющие глубокое обучение с подкреплением, модели среды, трансферное обучение и большие языковые модели. Такие подходы способны повысить адаптивность и универсальность интеллектуальных агентов. Дальнейшее развитие MARL связано с созданием более эффективных и масштабируемых многоагентных систем для работы в сложных динамических средах.

5. Заключение. В данной работе рассмотрена эволюция методов обучения с подкреплением от классических одноагентных алгоритмов к современным архитектурам глубокого и многоагентного обучения. Проведённый анализ показал, что развитие RL обусловлено необходимостью решения всё более сложных задач, связанных с увеличением размерности пространства состояний, ростом вычислительной сложности и переходом к коллективному взаимодействию интеллектуальных агентов.

В ходе исследования выполнено сравнение классических алгоритмов Q-learning и SARSA, которые сформировали теоретическую основу обучения с подкреплением. Показано, что Q-learning обеспечивает более высокую скорость поиска оптимальной стратегии, тогда как SARSA демонстрирует большую устойчивость при работе в условиях неопределённости и учитывает влияние исследовательского поведения агента. Вместе с тем использование табличного представления функций ценности существенно ограничивает применение данных методов в задачах с большим числом состояний и действий.

Рассмотрены современные методы глубокого обучения с подкреплением, включая алгоритмы DQN и PETS. Показано, что применение глубоких нейронных сетей и модельно-ориентированного прогнозирования позволяет эффективно работать со сложными высокоразмерными средами и значительно расширяет область практического применения RL. Однако данные методы характеризуются высокой вычислительной сложностью, значительными требованиями к объёму обучающих данных и необходимостью обеспечения устойчивости процесса обучения.

Особое внимание уделено переходу к многоагентному обучению с подкреплением. Рассмотрены особенности MARL, связанные с нестационарностью среды, экспоненциальным ростом пространства совместных действий, необходимостью координации поведения и ограниченной наблюдаемостью. Выполнен анализ архитектур DRON и парадигмы централизованного обучения с децентрализованным исполнением

(CTDE), реализованной в алгоритмах семейства MADDPG. Проведённое сравнение показало, что использование механизмов моделирования поведения других агентов и централизованного обучения способствует повышению эффективности коллективного принятия решений и улучшению адаптации к динамически изменяющимся условиям среды.

Проведённый сравнительный анализ различных классов алгоритмов показал, что развитие обучения с подкреплением характеризуется последовательным переходом от простых одноагентных моделей к сложным распределённым интеллектуальным системам. Классические методы остаются фундаментом современных алгоритмов, глубокое обучение обеспечивает работу с высокоразмерными данными, а многоагентные архитектуры позволяют решать задачи коллективного поведения и совместного принятия решений.

Таким образом, современное развитие обучения с подкреплением направлено не только на повышение эффективности отдельных алгоритмов, но и на создание адаптивных многоагентных интеллектуальных систем, способных функционировать в сложных динамических средах. Перспективными направлениями дальнейших исследований являются разработка масштабируемых архитектур MARL, совершенствование механизмов межагентной коммуникации, снижение вычислительных затрат обучения, а также интеграция методов глубокого обучения с подкреплением, моделей среды, трансферного обучения и больших языковых моделей для построения более универсальных интеллектуальных систем.

Список литературы

1. Саттон Р., Барто Э. Обучение с подкреплением: введение (Reinforcement Learning: An Introduction). – 2-е издание – Cambridge: MIT Press, 2018. – 552 с.
2. Уоткинс К., Дайан П. Q-learning // Machine Learning. 1992. Vol. 8. No. 3–4. P. 279–292.

3. Рамеш А., Нираджа С. Обучение на основе стратегии: алгоритм SARSA (On-Policy Temporal Difference Control Using SARSA) // Machine Learning. 1998. Vol. 32. No. 2. P. 157–183.
4. Мин В., ван Хасселт Х., Гез Д. и др. Глубокое обучение с подкреплением (Human-level Control through Deep Reinforcement Learning) // Nature. 2015. Vol. 518. No. 7540. P. 529–533.
5. Чуа К., Каллаган Р., Макаллистер Р., Левин С. Глубокое обучение с подкреплением на основе вероятностных ансамблей и выборки траекторий (Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models) // Advances in Neural Information Processing Systems. 2018. Vol. 31.
6. Хейр Т., Гупта А., Фуснер Л., Левин С. Temporal Difference Models: модельно-ориентированное глубокое обучение с подкреплением // Proceedings of the International Conference on Machine Learning. 2018. P. 2012–2021.
7. Хехе Ф., Бойд-Грабер Дж., Квок К. Моделирование поведения оппонентов для глубокого обучения с подкреплением (Deep Reinforcement Learning from Policy-Dependent Human Feedback) // Proceedings of the AAAI Conference on Artificial Intelligence. 2016.
8. Лоу Р., Ву И., Тамбе М. и др. Многоагентное актор-критик обучение для смешанных кооперативно-конкурентных сред (Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments) // Advances in Neural Information Processing Systems. 2017. Vol. 30.
9. Рашид Т., Самвелян М., Шредер К. и др. QMIX: монотонная факторизация функций ценности для глубокого многоагентного обучения с подкреплением // Proceedings of the International Conference on Machine Learning. 2018. P. 4295–4304.
10. Фёрстер Я., Фаркухар Г., Афурадис К. и др. Counterfactual Multi-Agent Policy Gradients // Proceedings of the AAAI Conference on Artificial Intelligence. 2018. Vol. 32. No. 1.

References

1. Sutton R., Barto A. Reinforcement Learning: An Introduction. 2nd ed. Cambridge: MIT Press, 2018. 552 p.
2. Watkins C., Dayan P. Q-learning // Machine Learning. 1992. Vol. 8. No. 3–4. P. 279–292.
3. Rummery G., Niranjan M. On-Policy Temporal Difference Control Using SARSA // Machine Learning. 1998. Vol. 32. No. 2. P. 157–183.
4. Mnih V., Hasselt H. van, Guez A. et al. Human-level Control through Deep Reinforcement Learning // Nature. 2015. Vol. 518. No. 7540. P. 529–533.
5. Chua K., Calandra R., McAllister R., Levine S. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models // Advances in Neural Information Processing Systems. 2018. Vol. 31.
6. Hare T., Gupta A., Foerster L., Levine S. Temporal Difference Models // Proceedings of the International Conference on Machine Learning. 2018. P. 2012–2021.
7. He H., Boyd-Graber J., Kwok K. Deep Reinforcement Learning with Opponent Modeling // Proceedings of the AAAI Conference on Artificial Intelligence. 2016.
8. Lowe R., Wu Y., Tamar A. et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments // Advances in Neural Information Processing Systems. 2017. Vol. 30.
9. Rashid T., Samvelyan M., Schroeder C. et al. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning // Proceedings of the International Conference on Machine Learning. 2018. P. 4295–4304.
10. Foerster J., Farquhar G., Afouras T. et al. Counterfactual Multi-Agent Policy Gradients // Proceedings of the AAAI Conference on Artificial Intelligence. 2018. Vol. 32. No. 1.