

УДК 004.8:004.946

## ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И ЕГО ПРИМЕНЕНИЕ ПРИ СОЗДАНИИ ИГРЫ «GOBBLET»

**Сикорская Галина Анатольевна**, доктор педагогических наук, доцент, доцент кафедры математики и цифровых технологий, Оренбургский государственный университет, Оренбург  
e-mail: galansik@mail.ru

**Носов Виталий Валерьевич**, кандидат физико-математических наук, доцент, доцент кафедры математики и цифровых технологий, Оренбургский государственный университет, Оренбург  
e-mail: puncker1978@mail.ru

**Кондауров Владислав Александрович**, студент, направление подготовки 02.03.01 Математика и компьютерные науки, Оренбургский государственный университет, Оренбург  
e-mail: vlad.kondourov@gmail.com

***Аннотация.** Использование обучения с подкреплением в играх связано с его потенциалом для исследования новых методов и алгоритмов машинного обучения. Игры предоставляют уникальные сценарии и среды, где можно исследовать и тестировать новые подходы RL и разрабатывать передовые модели и алгоритмы. Это способствует развитию области обучения с подкреплением в целом и содействует прогрессу в других областях искусственного интеллекта. В данной работе описывается и объясняется применение метода временной разницы (Temporal Difference, TD) в играх и обучении с подкреплением, а также выявлении его особенностей, отличающих его от других методов машинного обучения. Также представлена краткая история развития искусственного интеллекта, рассмотрены его основные виды и описано его применение в различных областях. Текст содержит пример использования искусственного интеллекта в игре «Gobblet».*

***Ключевые слова:** искусственный интеллект, история и виды искусственного интеллекта, применение искусственного интеллекта в играх.*

***Для цитирования:** Сикорская Г. А., Носов В. В., Кондауров В. А. Искусственный интеллект и его применение при создании игры «Gobblet» // Шаг в науку. – 2023. – № 4. – С. 84–90.*

## ARTIFICIAL INTELLIGENCE AND ITS APPLICATION IN CREATING THE GAME «GOBBLET»

**Sikorskaya Galina Anatolievna**, Doctor of Pedagogical Sciences, Associate Professor, Associate Professor of the Department of geometry and computer science, Orenburg State University, Orenburg  
e-mail: galansik@mail.ru

**Nosov Vitaly Valerievich**, Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of the Department of geometry and computer science, Orenburg State University, Orenburg  
e-mail: puncker1978@mail.ru

**Kondaurov Vladislav Alexandrovich**, student, training program 02.03.01 Mathematics and computer science, Orenburg State University, Orenburg  
e-mail: vlad.kondourov@gmail.com

***Abstract.** The use of reinforcement learning in games is due to its potential to explore new machine learning methods and algorithms. Games provide unique scenarios and environments where you can explore and test new RL approaches and develop advanced models and algorithms. This contributes to the development of the field of reinforcement learning in general and contributes to progress in other areas of artificial intelligence. This paper describes and explains the application of the Temporal Difference (TD) method in games and reinforcement learning, and also reveals its features*



that distinguish it from other machine learning methods. A brief history of the development of artificial intelligence is also presented, its main types are considered and its application in various fields is described. The text also contains an example of the use of artificial intelligence in the game “Gobblet”.

**Key words:** artificial intelligence, history and types of artificial intelligence, application of artificial intelligence in games.

**Cite as:** Sikorskaya, G. A., Nosov, V. V., Kondaurov, V. A. (2023) [Artificial intelligence and its application in creating the game «Gobblet»]. *Shag v nauku* [Step into science]. Vol. 4, pp. 84–90.

Искусственный интеллект (ИИ) – это область науки, которая занимается разработкой систем, способных анализировать данные и принимать решения, обладающие свойствами, ранее характерными только для человеческого интеллекта. Он становится все более неотъемлемой частью нашей жизни, и мы можем наблюдать все большее число проявлений ИИ в различных сферах деятельности человека.

История развития искусственного интеллекта включает в себя несколько этапов, каждый из которых привел к новым достижениям и открытиям в этой области.

Первый этап начался в 1950-х годах, когда ученые стали заниматься разработкой программ, способных обрабатывать информацию и выполнять задачи, доступные ранее только человеку. На этом этапе были созданы первые экспертные системы, делающие выводы на основе базы знаний, а также языки программирования, позволяющие создавать более сложные ИИ-системы. Однако в конце 1970-х годов пришел первый «AI winter» (затянувшаяся зима искусственного интеллекта), когда экспертные системы оказались неспособными заменить человеческий интеллект и многие компании прекратили финансирование исследований в области ИИ.

В 1980-х годах, благодаря новым подходам, таким как нейронные сети и генетические алгоритмы, интерес к искусственному интеллекту стал возрастать, что ознаменовало начало второго этапа его развития. На этом этапе ученые начали использовать новые методы машинного обучения, позволявшие создавать более сложные ИИ-системы, обучающиеся на основе большого количества данных, и обладающих способностью адаптации к изменяющейся среде. Однако в конце 1980-х годов наступил второй «AI winter», некоторые проекты оказались неудачными и финансирование снизилось.

Но в 2000-х годах ситуация изменилась, искусственный интеллект стал все более популярным, благодаря успехам в области машинного обучения. В 2010-х годах ученые начали использовать глубокое обучение, что привело к созданию ИИ-систем, способных распознавать объекты на изображениях, обрабатывать естественный язык и даже играть в компьютерные игры на уровне профессионалов.

Искусственный интеллект обладает множеством применений, включая медицину, финансы, автономные транспортные средства, робототехнику и многие другие области [4]. Обсудим различные виды ИИ и технологии, которые лежат в их основе, такие как машинное обучение, глубокое обучение и обучение с подкреплением.

На текущий момент мы можем выделить несколько основных областей, в которых используется ИИ.

Первая область – компьютерное зрение. Технологии искусственного интеллекта позволяют компьютерам обрабатывать изображения, распознавать объекты на фото или видео, и даже делать прогнозы о том, какие объекты окажутся на изображениях в будущем. Эта технология используется в таких областях, как безопасность, медицина, автомобильная промышленность и т. д.

Вторая область – естественный язык. Технологии искусственного интеллекта позволяют компьютерам понимать и генерировать естественный язык, что очень важно для машинного перевода, обработки больших объемов текстов, разработки голосовых помощников.

Третья область применения ИИ – робототехника. ИИ позволяет создавать роботов, которые могут работать в опасных условиях, непригодных для жизни человека, например, на дне океанов, на вершинах гор или в космосе. Также ИИ применяется в производстве роботов, выполняющих рутинные операции в промышленности.

Четвертая область применения ИИ – системы управления. Искусственный интеллект позволяет создавать системы управления, которые могут быстро реагировать на изменения внешней среды и принимать оптимальные решения на основе анализа данных. Эта технология используется в таких областях, как управление транспортом, производством, энергосистемами и т. д. [7].

Кроме того, искусственный интеллект широко используется в игровой индустрии, в частности, при создании компьютерных игр. Он может быть использован, как для управления поведением врагов и союзников, для создания ботов, играющих вместо людей, так и для управления игровой физикой и окружающей средой. Искусственный интеллект также применим

для создания игр, обладающих свойствами адаптации к игрокам, то есть, меняясь в зависимости от их действий, обеспечивая при этом более интерактивный игровой процесс [6].

Игровой искусственный интеллект имеет множество применений не только в индустрии компьютерных игр, он также может использоваться и в таких областях, как образование и медицина. С помощью ИИ, в частности, создаются обучающие игры для улучшения образования и симуляторы для обучения медицинским процедурам.

Мы рассмотрели только некоторые из областей применения технологии искусственного интеллекта. В целом, ИИ является одной из самых быстро развивающихся технологий в мире, и, очевидно, в будущем его применение станет еще более широким и разнообразным.

Остановимся на одном из популярных методов искусственного интеллекта, машинном обучении.

Машинное обучение – технология, позволяющая компьютерным системам обучаться на основе данных, без явного программирования, что позволяет компьютеру автоматически улучшать свою производительность в решении определенной задачи, используя опыт, накопленный при работе с данными [1]. Машинное обучение используется во многих сферах, включая обработку естественного языка, компьютерное зрение, рекомендательные системы, анализ данных и многое другое. В играх машинное обучение может быть использовано для создания ИИ оппонентов, с целью адаптации и улучшения своих навыков на основе игрового опыта<sup>1</sup>.

Основными подходами к машинному обучению являются обучение с учителем, обучение без учителя и обучение с подкреплением.

В обучении с учителем компьютер обучается на основе маркированных данных, где каждый элемент данных имеет соответствующую метку, указывающую на правильный ответ. В обучении без учителя, компьютер ищет закономерности и структуры в немаркированных данных. При обучении же с подкреплением, компьютер обучается, играя сам с собой, получая награду или штраф, в зависимости от успешности в процессе решения поставленной задачи.

В 1960-х годах американский психолог Б. Ф. Скиннер провел серию экспериментов с кошкой, использовавшиеся для создания метода обучения с подкреплением на основе награды и наказания. В результате экспериментов Скиннер показал, что животные и люди лучше осваивают новые навыки, если за правильное

поведение получают награду [2]. В 1984 году Ричард Саттон и Эндрю Барто разработали метод обучения с подкреплением на основе полученных Скиннером результатов. Этот метод использовался для обучения агентов в динамических и неопределенных средах и стал одним из первых методов машинного обучения, используемых в искусственном интеллекте и робототехнике [8]. Сегодня метод обучения с подкреплением широко используется для создания агентов, способных самостоятельно принимать решения на основе наград и штрафов. Этот подход успешно применяется в различных областях, таких как автономная навигация, управление роботами, а также в игровых приложениях. Суть метода заключается в том, что агент находится в некотором состоянии среды и принимает решение о том, какое действие следует совершить. За это действие агент получает награду или штраф. Задача агента состоит в максимизации получаемой награды. Для решения этой задачи обучения с подкреплением используются различные алгоритмы, такие как Q-обучение, SARSA и др., основанные на идее оценки качества действий в заданном состоянии и выбора действия с максимальной оценкой.

Обучение с подкреплением (рисунок 1) широко применяется в игровых приложениях для создания игровых ботов, которые могут соревноваться с людьми в сложных стратегических играх, таких как шахматы, го или покер. Кроме того, этот подход используется в управлении беспилотными автомобилями и дронами, с целью оптимальной реализации поставленных задач.

Обучение с подкреплением часто использует метод временной разности – Temporal Difference (TD) (рисунок 2). Одной из ключевых особенностей алгоритма TD является его способность к обучению в режиме онлайн, то есть на каждом шаге взаимодействия агента со средой, в отличие от более традиционного метода, который требует обработки всего набора данных перед началом обучения. В алгоритме TD функция ценности состояний определяется путем сравнения текущей оценки со следующей на основе получаемой награды, и, соответственно, оценки состояний, полученных в результате перехода из текущего состояния в следующее.

В частности, TD нашел широкое применение в игровой индустрии для создания AI-игроков [3]. Например в шахматах, где действия агента влияют на будущее состояние и выигрыш, метод Монте-Карло требует генерации и анализа большого количества случайных партий. Это может быть вычислительно затратным и время затратным. В то время как TD обновляет

<sup>1</sup> Sammut C., Webb G. I. (2010) Encyclopedia of Machine Learning: 1st ed. NY: Springer New York, 1031 p. <https://doi.org/10.1007/978-0-387-30164-8>.

оценки ценности состояний на основе текущего опыта и может достигать оптимальных стратегий даже с ограниченным числом партий. Генетические алгоритмы могут быть эффективными, но они обычно требуют большого числа попыток и эволюционных итераций, что может быть время затратным. TD обновляет

оценки ценности действий на основе текущего опыта и может сходиться к оптимальной стратегии быстрее. Таким образом, TD может быть предпочтительным в сценариях, где ограничены ресурсы по вычислениям и доступному опыту, и когда необходима быстрая адаптация к изменяющейся среде.

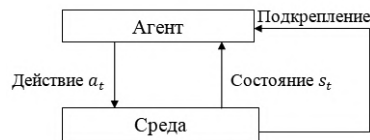


Рисунок 1. Алгоритм обучения с подкреплением  
 Источник: заимствовано из работы [5]

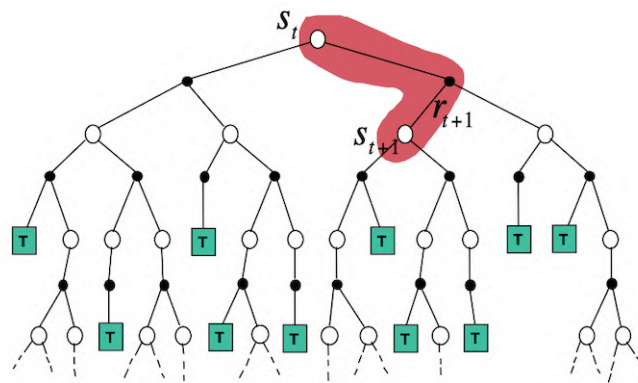


Рисунок 2. Алгоритм Temporal Difference  
 Источник: заимствовано из работы [9]

Целью данного исследования является применение метода временной разницы (TD) для улучшения искусственного интеллекта в игре «Gobblet», с целью улучшения игрового опыта и повышения интеллектуальной способности игрового агента.

Для достижения цели и задач исследования, включая использование метода временной разницы для применения искусственного интеллекта в игре «Gobblet», использовались следующие инструменты и технологии:

Python – это высокоуровневый язык программирования, который широко используется в машинном обучении и исследованиях в области искусственного интеллекта. Он предоставляет множество библиотек и инструментов для реализации алгоритмов машинного обучения, включая TensorFlow.

PyCharm – это интегрированная среда разработки (IDE) для Python, которая предоставляет множество удобных инструментов для разработки, отладки и анализа кода. Он помогает разработчикам эффективно со-

здавать и тестировать программное обеспечение.

QT Designer – это инструмент для создания графического интерфейса пользователя (GUI) в приложениях, написанных на языке программирования Python с использованием библиотеки PyQt. Он позволяет легко создавать интерфейсы для приложений, включая игровые интерфейсы.

TensorFlow – это библиотека машинного обучения и глубокого обучения, разработанная Google. Она предоставляет множество инструментов и ресурсов для создания, обучения и развертывания моделей искусственного интеллекта. TensorFlow может использоваться для реализации алгоритмов обучения с подкреплением, включая TD, и для создания игровых агентов.

С использованием этих инструментов и технологий был реализован алгоритм TD для игры «Gobblet» с целью улучшения игрового опыта и повышения интеллектуальной способности игрового агента, а также создан соответствующий графический интерфейс с помощью QT Designer для визуализации результатов

и взаимодействия с игрой.

Для решения задачи применения искусственного интеллекта при разработке компьютерных игр необходима следующая архитектура решения:

*Сбор данных и предобработка:*

Сначала необходимо собрать данные, которые будут использоваться для обучения и оценки моделей ИИ и МО. Это могут быть игровые данные, действия игроков, состояния игровых миров и другие параметры игры.

Далее проводится предобработка данных, включая их очистку, масштабирование и преобразование в удобный формат для дальнейшей обработки.

*Выбор моделей и методов:*

Выбираются модели машинного обучения или искусственного интеллекта, которые наилучшим образом подходят для конкретной игры и её задач. Это могут быть методы обучения с подкреплением, нейронные сети, алгоритмы генетического программирования и многие другие.

Определяются методы обучения, функции потерь и метрики, которые будут использоваться для оценки производительности моделей.

*Обучение и тестирование:*

Модели обучаются на подготовленных данных, но важно отметить, что в некоторых случаях среда может быть динамической и меняться со временем. В таких ситуациях модели могут обучаться в реальном времени, взаимодействуя с окружающей средой и принимая решения на основе непосредственного опыта. Этот процесс, известный как обучение с подкреплением в реальном времени, позволяет моделям адаптироваться к изменяющейся среде, даже если исходные данные о среде ограничены или отсутствуют.

*Интеграция в игру:*

Обученные модели и алгоритмы интегрируются в среду, чтобы они могли взаимодействовать с игрой и принимать решения на основе текущей игровой ситуации.

Разрабатывается интерфейс взаимодействия между моделями ИИ и игрой, что позволяет им взаимодействовать с игровым миром.

*Оптимизация и настройка:*

Проводится оптимизация моделей и алгоритмов для обеспечения их эффективной работы в реальном времени.

Производится настройка параметров моделей с целью достижения желаемых игровых результатов и баланса между сложностью игры и уровнем удовольствия для игроков.

*Тестирование и доработка:*

Модели и игровой процесс тестируются на реальных игроках для оценки их производительности

и адаптации к действиям игроков. В случае необходимости проводятся доработки и улучшения моделей и игрового опыта.

В игре «Gobblet» используется нейросеть с несколькими скрытыми слоями для аппроксимации Q-функции. Нейросеть принимает состояние среды (размером 4x12, то есть 48 нейронов) и имеет два скрытых полносвязных слоя по 64 нейрона каждый с функцией активации ReLU. Это позволяет модели обучаться сложным нелинейным зависимостям в данных. Выходной слой использует линейную функцию активации, чтобы предсказать Q-значения для каждого действия.

Для обучения используется оптимизатор Adam, который обновляет веса модели на основе градиентов функции потерь. Этот оптимизатор обладает высокой скоростью сходимости и адаптивной настройкой скорости обучения.

Функция потерь Huber используется для измерения расхождения между предсказанными и целевыми Q-значениями. Huber Loss более устойчива к выбросам и может быть полезна в задачах, где данные содержат значительные колебания.

Для балансировки исследования и эксплуатации применяется эpsilon-стратегия. Значение  $\epsilon$  определяет вероятность случайного выбора действия, а  $(1 - \epsilon)$  определяет вероятность выбора действия с наибольшим значением Q-функции. Такие архитектурные и методические решения помогают модели эффективно обучаться и принимать обоснованные решения в среде. Далее, с помощью кода мы регулируем баланс между исследованием и эксплуатацией в процессе обучения агента, используя Q-функции. Вот краткое описание основных шагов:

Уменьшение `self.epsilon` – это параметр, определяющий вероятность совершения исследовательских действий (случайных действий) против эксплуатации (выбора действий на основе Q-значений). После каждого обновления состояния среды, значение `self.epsilon` умножается на `self.epsilon_decay`, чтобы постепенно снижать вероятность случайных действий.

Создание переменной `target`: создается переменная `target`, которая содержит предсказанные значения Q-функции для текущего состояния `state`.

Обновление целевых Q-значений: этот шаг играет важную роль в процессе обучения агента, особенно в контексте ошибки TD (оценки разницы во времени). Целевые Q-значения обновляются в зависимости от того, завершена ли игра (`done`). Если игра завершилась, значение целевого Q-значения для выбранного действия устанавливается равным полученной награде. В противном случае, вычисляется максимальное предсказанное Q-значение для следующего состоя-

ния, и целевое Q-значение для выбранного действия вычисляется как сумма награды и дисконтированной будущей максимальной ожидаемой награды.

TD-ошибки используются внутри методов `self.model.predict` и `self.target_model.predict` для оценки Q-значений и обновления целевых Q-значений. Важно

отметить, что TD-методы обновления Q-значений, такие как Q-learning и SARSA, основаны на оценке TD-ошибок. Ошибка TD для конкретного состояния и действия представляет собой разницу между текущей оценкой Q-значения и целевым Q-значением:

$$TD\_error = Q\_target - Q\_current,$$

где:

$Q\_target$  – целевое Q-значение, вычисленное на основе будущих состояний и награды;

$Q\_current$  – текущая оценка Q-значения для данного состояния и действия.

Таким образом, TD-методы используются для вычисления ошибок прогнозирования и обновления Q-значений на основе этих ошибок в контексте обучения с подкреплением.

Обучение нейросети: нейросеть обучается на основе целевых Q-значений с целью минимизации функции потерь. Градиент функции потерь по отношению к переменным модели используется для обновления весов нейросети.

Выполнение действия в среде: агент выполняет действие в среде в соответствии с выбором, основанным на Q-значениях или случайным выбором. Награда (положительная или отрицательная) зависит от результата действия.

Этот процесс повторяется в течение обучения, и агент взаимодействует со средой, обновляя свою модель на основе полученного опыта.

В результате обучения с использованием метода обновления Q-значений с учетом ошибки TD, мы получаем нейросеть, способную играть в заданную игру самостоятельно. Она учится корректировать свои действия на основе опыта и наград, стремясь максимизировать свою накопленную награду. Этот метод обучения с подкреплением позволяет нейросети улучшать свои стратегии и принимать решения, оптимизированные с точки зрения цели задачи, в которой она обучалась.

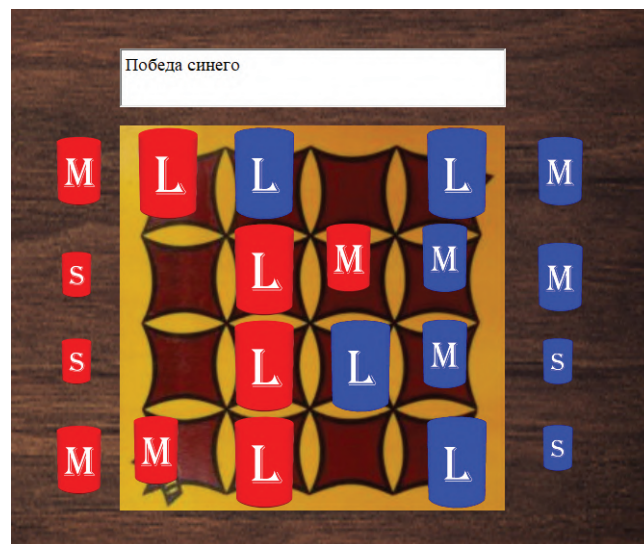


Рисунок 3. Отображение победы синего игрока

Источник: скриншот готовой игры

### Литература

1. Джонс М. Т. Программирование искусственного интеллекта в приложениях / пер. с англ. А. И. Осипов. – М.: ДМК Пресс, 2018. – 311 с.
2. Сикорская Г. А., Кондауров В. А. О технологиях искусственного интеллекта // Университетский ком-

плекс как региональный центр образования, науки и культуры: материалы Всерос. науч.-метод. конф., Оренбург, 26–27 янв. 2023 г. – Оренбург: ОГУ, 2023. – С. 1379–1385.

3. Сикорская Г. А., Ситников В. А. Об эволюции искусственного интеллекта в видеоиграх // Университетский комплекс как региональный центр образования, науки и культуры: материалы Всерос. науч.-метод. конф. (с междунар. участием), Оренбург, 25–27 янв. 2021 г. – Оренбург: ОГУ, 2021. – С. 1686–1689.

4. Солнцева О. Г. Аспекты применения технологий искусственного интеллекта // E-Management. – 2018. – Т. 1. – № 1. – С. 43–51. – <https://doi.org/10.26425/2658-3445-2018-1-43-51>.

5. Станкевич Л. А. Интеллектуальные системы и технологии : учебник и практикум для вузов / Л. А. Станкевич. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2023. – 495 с.

6. Шампандар А. Д. Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия / пер. с англ. К. А. Птицына, Н. А. Шпака. – М.: ООО «И.Д. Вильямс», 2007. – 768 с.

7. Mykel J. K. (2015) Decision Making Under Uncertainty: Theory and Application. – MIT Lincoln Laboratory Series, 349 p.

8. Sutton R. S., Barto A. G. (2018) Reinforcement Learning, second edition: An Introduction. London: The MIT Press, 552 p.

9. Yann D. (2019) Temporal difference learning GitHub Pages – URL: <https://yanndubs.github.io/machine-learning-glossary/reinforcement/tdl> (accessed 14.03.2023).

Статья поступила в редакцию: 28.04.2023; принята в печать: 20.11.2023.

Авторы прочитали и одобрили окончательный вариант рукописи.