МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

СИМУЛЯЦИЯ ПОВЕДЕНИЯ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Юдина Павла Владимировича

Научный руководитель
доцент, к. п. н. _____ В. А. Векслер

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Симуляция поведения мультиагентных систем является актуальной темой для современного мира. Симуляция помогает в случаях, когда требуется найти наилучший алгоритм для поведения и применить этот подход в реальной ситуации. Также с помощью симуляции можно предсказать поведение агентов при некоторых заданных параметрах. В симуляции отлично помогают алгоритмы обучения с подкреплением, которые позволяют научить агентов действовать так, как будет выгодно действовать в том состоянии и в той ситуации, в которой они находятся.

Для мониторинга и анализа результатов результатов, помимо выходных данных, зачастую необходима и графическая визуализация. Поэтому в данной работе также будет разработана визуализация процессов игрового мира.

Также сейчас всё большую популярность обретают приложения с простым интерфейсом, который будет доступен на всех устройствах. Поэтому было принято решение разработать Telegram-бота, который позволит задавать параметры для обучения и получать видео с симуляцией и графики для анализа.

Актуальность работы заключается в том, что данные, полученные в результате, могут помочь в улучшении некоторых существующих систем, либо же помогут разработать новые. При этом эти системы могут быть как из производственной области, так и из области разработки компьютерных игр.

Целью работы является исследование и сравнение алгоритмов обучения с подкреплением: q-learning и SARSA.

Поставлены следующие задачи:

— реализовать модель симуляции поведения мультиагентной среды, используя футбольный матч как предметную область;
— обучить агентов (футболистов), используя алгоритмы Q-learning и SARSA;
— реализовать визуализацию симуляции;
— реализовать Telegram-бот для тестирования работы алгоритмов и вывода графиков.

В данной работе используются следующие инструменты и технологии:

— язык программирования Python 3;
— библиотека для разработки и сравнения алгоритмов обучения с подкреплением – OpenAI gym;
— библиотека pygame для визуализации симуляции;

— библиотека pyTelegramBotAPI для реализации Telegram бота;
— по для автоматизации развертывания приложений Docker;
— сервис для управления образами и контейнерами Yandex Container Registry.

1 Анализ предметной области

В условиях современного мира, когда повышается сложность информации и событий, простые методы анализа могут быть неэффективными для моделирования систем. Поэтому эти ситуации требуют совершенно иных, более гибких методов моделирования и обработки. Одним из таких методов считаются мультиагентные системы. К примеру, мультиагентные системы могут использоваться для моделирования ситуаций на предприятиях, в торговле, либо же в науке.

Сейчас становится более актуальной разработка мультиагентных систем и сравнительный анализ алгоритмов обучения с подкреплением. На основе этой работы можно будет сделать вывод о том, какие аспекты для создания мультиагентной системы нужно учесть и какой алгоритм будет более эффективным.

В данной работе в качестве системы для реализации был выбран футбольный матч. В этой системе средой является футбольное поле вместе с набором правил, а агентами являются футболисты. Этот выбор обусловлен тем, что футбол является наиболее популярным видом спорта в настоящий момент, поэтому моделирование этой системы будет более актуальной, а также будет больше возможностей для проверки и анализа алгоритмов.

Для создании успешной мультиагентной системы необходимо:

— разработать среду, наиболее приближенную к реальной жизни;
— описать и внедрить свод правил, действующих в этой среде;
— разработать описание агентов, которое будет приближено к реальной жизни и внедрить это описание в систему.

Но для того чтобы система могла наиболее эффективно проявить себя, было принято решение использовать два алгоритма машинного обучения с подкреплением: q-learning и SARSA. На основе сравнительного анализа обоих алгоритмов будет сделан вывод о том, какой алгоритм будет эффективен для рассматриваемой системы.

Также для того, чтобы можно было пронаблюдать за процессом симуляции, необходимо разработать визуализацию системы. Это позволит более наглядно понять, какие действия выполняют футболисты при обучении.

Для более удобного и быстрого обучения нужно разработать такой интерфейс, который позволит любому пользователю задать параметры для обучения и выбрать модель, которая будет обучена.

2 Разработка модели симуляции

2.1 Среда и агенты

Среда – это некоторое окружение, которое существует по заранее созданному набору правил и взаимодействует с агентами. Агентом может являться всё, что имеет возможность взаимодействовать со средой, используя датчики, которые обрабатывают и воспринимают состояния среды, и исполнительные механизмы, которые действуют на среду и изменяют её [1].

Схема взаимодействия среды представлена на (Рис. 1)

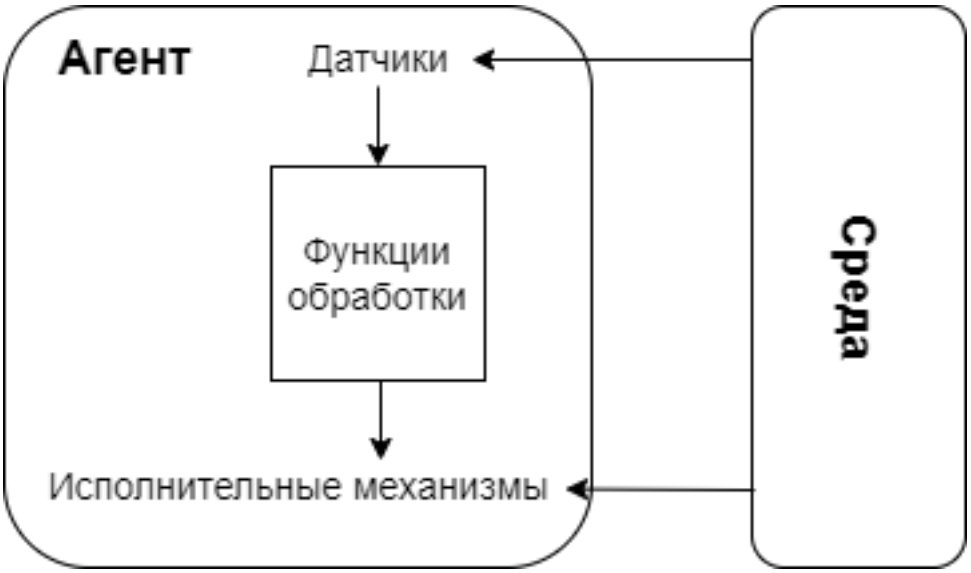

Рисунок 1 – Схема взаимодействия среды

В данной работе была выбрана среда на основе футбольного матча. Все правила основываются на существующих футбольных правилах.

Конечной целью каждой из команд является победа в матче, т.е. необходимо забить больше голов, чем соперник. У каждого агента есть набор атрибутов, например: координаты в настоящий момент времени, координаты в предыдущий момент времени, цвет, сторона, и т.д. В каждой среде должен быть свой свод правил. Эти правила делают среду уникальной, а также ограничивают её и улучшают результаты симуляции.

2.2 Состояния

2.2.1 Состояния агентов

Для успешно функционирующей системы был рассмотрен и формализован общий набор состояний, в которых могут находиться агенты в определённый момент времени.

2.2.2 Состояния среды

Также, для того, чтобы можно было более точно симулировать модель футбольного матча, используются состояния среды, которые помогут отследить и определить, в какой момент необходимо применять обучение.

2.3 Действия

Также, наряду с состояниями агентов, необходимо описать действия агентов, которые будут совершаться ими в определенный момент времени.

Обработку действий агента можно разделить на несколько этапов:

- изменение переменной события;
- изменение переменной среды на «Действие»;
- пересчёт координат агента;
- пересчёт координат мяча (при необходимости);
- изменение второстепенных переменных (например, скорость мяча, либо переменную-проверку для агента-владельца мяча).

2.4 Реализация среды

В библиотеке OpenAI gym существует родительский класс Env. Этот класс содержит базовые описания методов, необходимые для работы с окружением [2].

Основные методы класса Env:

- step – вызывается при необходимости перехода к следующему шагу;
- reset – вызывается при необходимости «сбросить» окружение;
- render – вызывается при необходимости визуализации окружения.

Все эти методы были переопределены в ходе разработки.

2.4.1 Шаг среды

Каждый следующий шаг позволяет обработать следующее состояние среды и вернуть информацию о ней. При симуляции один шаг приравнивается к одному моменту времени. При выполнении шага пересчитываются позиции игроков, при необходимости меняется состояние и вызывается функция, в которой определяются дальнейшие действия футболистов.

2.4.2 Создание экземпляра среды

Для инициализации команд возможно выбрать одну из нескольких тактических схем и один из стилей игры.

3 Обучение агентов

3.1 Обучение с подкреплением

Обучение с подкреплением (reinforcement learning) – это набор методов машинного обучения, при котором само обучение происходит методом проб и ошибок. Агент, взаимодействуя со средой, изменяет её и получает награду за успешные действия, либо получает штраф за негативные действия. Принцип получения награды при успешных действиях и называется подкреплением [3].

Наиболее популярными областями, в которых используется обучение с подкреплением являются:

- компьютерные игры;
- робототехника;
- системы управления ресурсами;
- системы, предоставляющие индивидуальную рекомендацию.

Для лучшей симуляции поведения агентов необходимо научить их справляться с проблемами, которые возникают на их пути. Поэтому было принято решение использовать методы машинного обучения. Для реализации были выбраны методы q-learning и SARSA [4].

3.2 Q-learning

Q-обучение – это метод машинного обучения с подкреплением, который применяется при агентном подходе. Общий принцип алгоритма следующий: на основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой [5].

Одно из преимуществ Q-обучения — то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений [6].

3.3 SARSA

SARSA (State-Action-Reward-State-Action)– это алгоритм изучения марковской политики процесса принятия решений, используемый в области обучения с подкреплением. Алгоритм похож на Q-learning, но отличие в том, что

значения Q рассчитываются на основе действий, выполняемых текущей политикой. Это означает, что SARSA является более гибким алгоритмом. Алгоритмы, использующие одну политику для оценки значения полезности Q и генерации следующего действия, называются алгоритмами на основе политики «on-policy» [7].

3.4 Сравнение алгоритмов

Для того, чтобы сравнить алгоритмы между собой, было принято решение обучить несколько моделей и сравнить результаты.

3.4.1 Обучение модели на основе алгоритма q-learning

В качестве основы используется среда, в которой присутствуют только игроки одной команды, которые учатся забивать мяч в ворота. При этом в воротах противоположной команды находится вратарь. Для обучения использовались гиперпараметры α и γ, которые равны 0.1 и 0.6, соответственно. Процесс обучения происходит следующим образом: в самом начале среда находится в начальном состоянии, при котором мяч находится в центре, а футболисты находятся на своей исходной позиции. Далее симулируется матч до того момента, пока не будет забит гол. В этом случае среда переходит к начальному состоянию. На основе этого проводятся эксперименты.

3.4.2 Обучение модели на основе алгоритма SARSA

По аналогии с предыдущей моделью, в качестве основы для среды используются игроки одной команды и вратарь из противоположной. К гиперпараметрам добавилось значение ϵ, равное 0.1. Процесс обучения аналогичен тому, что был представлен выше. Результат экспериментов, к правило, показывает о том, что алгоритм q-learning работает лучше для выбранной среды.

3.4.3 Обучение моделей при игре друг против друга

Для того чтобы лучше рассмотреть эффективность обоих алгоритмов, была обучена модель в среде, в которой существуют все двадцать два агента, задача которых забить мяч в ворота противоположной команды. На основе этого также были получены результаты, которые подтверждают, что для выбранной среды более эффективен алгоритм q-learning.

4 Реализация приложения для тестирования

Для тестирования алгоритмов было разработано приложение, в котором можно обучить модель, увидеть симуляцию матча и проанализировать графики, полученные после обучения. Схему приложения можно увидеть на (Рис. 2)

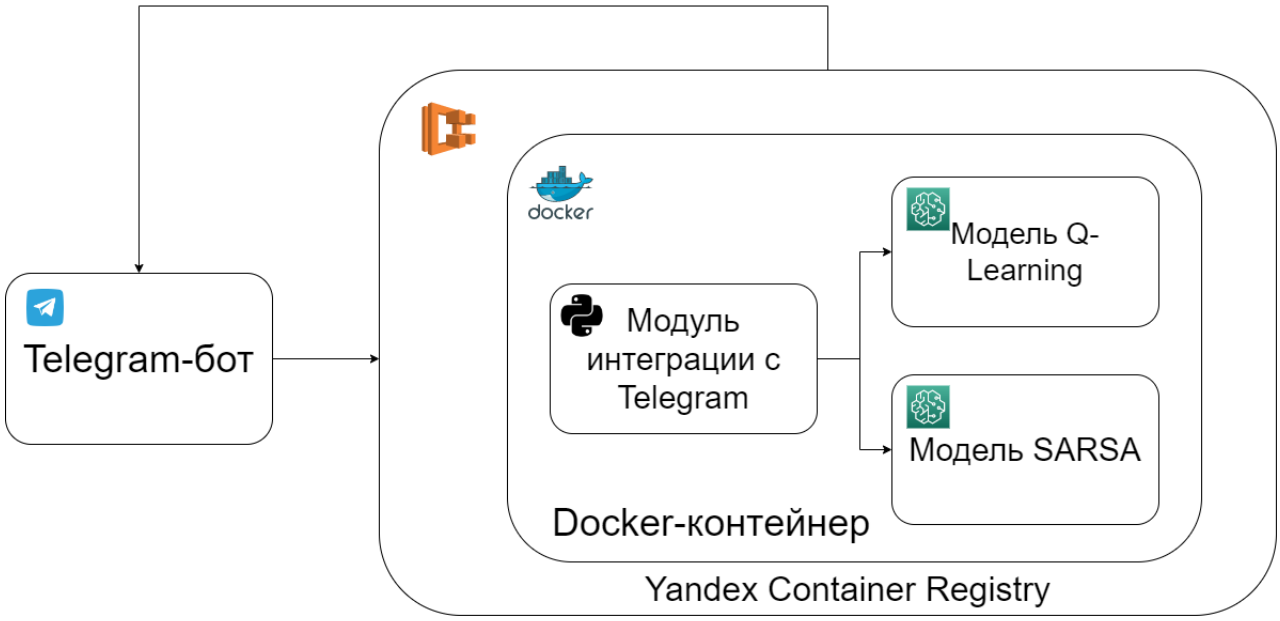

Рисунок 2 – Схема приложения

4.1 Визуализация матча

Визуализация симуляции выполнена с помощью библиотеки pygame [8]. Pygame — это набор модулей языка программирования Python, предназначенный для написания компьютерных игр и мультимедиа-приложений [9]. Принцип отрисовки следующий: при инициализации создается окно, а в каждый момент времени отрисовывается среда (футбольное поле) и все агенты (игроки команд).

4.2 Разработка бота

Для разработки бота Telegram предоставляет собственный API, используя который, можно написать настраиваемый клиент для бота [10]. В данной работе была выбрана реализация API – библиотека pyTelegramBotAPI. Это библиотека для языка программирования python, которая предоставляет функционал для реализации собственного бота.

4.3 Развёртывание приложения

4.3.1 Docker

Docker – это платформа, которая позволяет доставлять и запускать приложения. Docker помогает создавать контейнеры, а также автоматизировать их запуск и развёртывание. Также Docker помогает управлять жизненным циклом приложения [11].

Контейнеризация – это способ упаковки приложения вместе со всеми его зависимостями.

Docker-образ – это шаблон, который позволяет создавать Docker-контейнеры. Образ представляет собой пакет, или же сценарий, который содержит необходимые элементы предложения, например код, команды для импорта библиотек, создание окружения [12].

Docker значительно упрощает тестирование приложения и управление им. Поэтому для данной работы был выбран именно этот способ развёртывания.

Для полноценного тестирования необходимы три составляющие:

— dockerfile, в котором описаны команды для сборки образа;
— docker-compose файл, в котором описаны параметры для поднятия Docker-контейнера;
— скрипт, позволяющий автоматизировать сборку образа и поднятие контейнера.

4.3.2 Yandex Cloud

Для того чтобы приложение работало полностью автономно и было более надежным, было принято решение использовать Yandex Cloud Registry. Этот сервис позволяет хранить образы и управлять ими [13].

ЗАКЛЮЧЕНИЕ

Целью работы являлось исследование и сравнение алгоритмов обучения с подкреплением: q-learning и SARSA.

Для выполнения данной работы необходимо было создать среду, в которой будут обучаться агенты, а также обучить агентов в этой среде, используя данные алгоритмы. На основе полученных данных об обучении удалось сделать вывод об эффективности этих алгоритмов для рассматриваемой среды.

В ходе разработки и реализации проекта была полностью достигнута цель, поставленная в начале работы.

Были выполнены следующие задачи:

- реализована модель симуляции поведения мультиагентной среды, используя футбольный матч как предметную область;
- обучены агенты (футболисты), используя алгоритмы Q-learning и SARSA;
- реализована визуализация симуляции;
- реализован Telegram-бот для тестирования работы алгоритмов и вывода графиков.

Данная работа позволит помочь в улучшении существующих систем моделирования, а также в создании новых систем, в которые можно будет внедрить машинное обучение с подкреплением.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Девятков, В. В.* Системы искусственного интеллекта / В. В. Девятков. — Москва: Издательство МГТУ имени Н.Э. Баумана, 2001.

2 Api Gym [Электронный ресурс]. — URL: https://www.gymlibrary.ml/content/api/ (Дата обращения 30.05.2022). Загл. с экр. Яз. анг.

3 *Graesser, L.* Foundations of Deep Reinforcement Learning: Theory and Practice in Python / L. Graesser. — Boston: Addison-Wesley Professional, 2019.

4 *Ravichandiran, S.* Hands-On Reinforcement Learning with Python / S. Ravichandiran. — Birmingem: Packt Publishing, 2018.

5 *Розенблатт, Ф.* Принципы нейродинамики: Перцептроны и теория механизмов мозга / Ф. Розенблатт. — Москва: Мир, 1965.

6 *Watkins, C.* Learning from Delayed Rewards / C. Watkins. — Cambridge, 1989.

7 *Rummery, G.* Online Q-learning using connectionist systems / G. Rummery. — Cambridge, 1994.

8 *McGugan, W.* Beginning Python Games Development / W. McGugan. — New York: Apress, 2015.

9 Pyagame intro – pygame [Электронный ресурс]. — URL: https://www.pygame.org/docs/tut/PygameIntro.html (Дата обращения 30.05.2022). Загл. с экр. Яз. анг.

10 Telegram API – Methods [Электронный ресурс]. — URL: https://core.telegram.org/methods (Дата обращения 30.05.2022). Загл. с экр. Яз. анг.

11 *Goasguen, S.* Docker Cookbook / S. Goasguen. — Sebastopol: O'Reilly Media, 2015.

12 *Моуэт, Э.* Использование Docker / Э. Моуэт. — Sebastopol: O'Reilly Media, 2017.

13 Yandex Container Registry [Электронный ресурс]. — URL: https://cloud.yandex.ru/docs/container-registry/ (Дата обращения 30.05.2022). Загл. с экр. Яз. рус.