МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ НАРУШИТЕЛЯ ПО ПРОНИКНОВЕНИЮ НА ОХРАНЯЕМЫЙ ОБЪЕКТ НА ОСНОВЕ МЕТОДОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Федоров Кирилл Александрович

Гродненский государственный университет им.Янки Купалы,
факультет математики и информатики,
магистрант 1 курса специальности 1-53 80 01 "Автоматизация" с профилизацией "Анализ и управление в системах цифровой экономики"

Научный руководитель Кадан Александр Михайлович, к.т.н., доцент

В современном мире все больше возрастает необходимость в предсказании действий нарушителей, которые пытаются проникнуть на охраняемый объект. В связи с этим возрастает необходимость в программных продуктах, которые бы моделировали охраняемый объект и поведения нарушителей.

Задача заключается в том, чтобы реализовать систему моделирования поведения нарушителя для контроля проникновения на охраняемый объект с использованием обучение с подкреплением. Это должно позволить решить проблему автоматического обучения, и поиска оптимальных решений, которые не теряют актуальность с течением времени.

Ключевые слова: обучение с подкреплением, многоагентные системы, физические двигатели, библиотеки глубокого обучения, модельное обучение.

Для выполнения поставленной задачи необходимо применить многоагентные системы. Система с многими агентами — это технология с применением агентов, в которой группа автономных агентов взаимодействуют в среде для достижения общей цели. Взаимодействие осуществляется путем сотрудничества или конкуренции, обмена или не обмена знаниями [1]. Многоагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы. Обычно в многоагентных системах исследуются программные агенты. Тем не менее, составляющими системы с многими агентами могут также быть роботы, люди или команды людей. Также, такие системы могут содержать и смешанные команды. В многоагентных системах может проявляться самоорганизация и сложное поведение даже если стратегия поведения каждого агента достаточно проста. Это лежит в основе так называемого роевого интеллекта.

Важно также и то, что обучение с подкреплением в системе с многими агентами может привести к выявлению ранее неизвестных стратегий действий. В частности, если рассматривать среду с двумя конкурирующими командами, система может сгенерировать противодействия данным действиям.

При моделировании предусматривается наличие двух команд одна команда нарушителей, а вторая команда стражей.

Задача нарушителей состоит в том, чтобы избегать прямой видимости со стражами, а стражей установить прямую видимость с злоумышленниками.

При моделировании предусматривается наличие объектов, которые разбросаны по всей среде. Агенты могут захватывать, а также заблокировать объекты на месте. Предусматривается также случайно сгенерированные неподвижные комнаты и стены, среди которых агенты должны научиться ориентироваться.

Перед началом игры нарушителям предоставляется подготовительная фаза, когда стражи обездвижены, давая возможность скрыться или изменить среду под свои нужды.

Нет явных стимулов для агентов взаимодействовать с объектами в окружающей среде, единственный стимул через цель команды. Агенты получают командное вознаграждение. Нарушители получают награду 1, если они все скрыты от стражей, и -1, если какой-либо страж видит любого из нарушителей. Стражи получают противоположную награду, -1, если все нарушители скрыты, и +1 в противном случае.

Для того чтобы ограничить поведение агентов в случае, если они выходят слишком далеко за пределы игровой площадки им дается отрицательная награда.

Поскольку агенты взаимодействуют с окружающей средой посредством контакта, точное и быстрое моделирование динамики контакта имеет решающее значение для управления агентами на основе моделей [2]. Естественно, проводить реалистичное и эффективное моделирование контактов для исследований позволяет успешно контролировать обучение.

Инструменты моделирования твердого цела

Существуют многие инструменты моделирования твердого тела, которые имитируют динамику контакта. Для исследователей сложно найти и выбрать лучшие симуляторы для своих задач среди множества вариантов.

Для работы необходимо обеспечить всестороннюю оценку точности и скорости моделирования контактов на наиболее широко используемых физических двигателях для различных ситуаций от систем с одним агентом с ограниченным числом контактов до сложных систем.

Тест проводился на следующих физических двигателях:

1. RaiSim
2. Bullet
3. ODE
4. MuJoCo
5. DART

В результате сравнения и тестирования разных физических двигателей было принято решения что в разработке приложения лучше использовать такие физические двигатели как Mujoco или DART. Исходя из тестов данные физические двигатели являются лучшими для разработки приложения, которое будет моделировать среду с многими агентами.

Также в качестве альтернативы можно использовать плагин Unity ML-Agents. Данный плагин предоставляет во смежность разработки как игр, так и агентов для них. Данный плагин предоставляет большие возможности для исследователей искусственного интеллекта.

Библиотеки глубокого обучения.

На данный момент разработано много библиотек глубокого обучения, и может быть трудно выбрать лучшую библиотеку. Поэтому нужно сравнить различные библиотеки.

Чтобы выбрать библиотеку, определим некоторые критерии, которые являются наиболее важными:

1. Реализованные современные алгоритмы.
2. Хорошая документация / учебные пособия и примеры.
3. Разборчивый код, который легко изменить.
4. Регулярные обновления и активное сообщество.
5. Tensorboard поддерживается.
6. Наличие других функций (например, векторизованных сред)

Векторизованная среда - это метод обучения с многими процессами; вместо обучения нашего агента в одной среде, мы обучаем его в n средах(потому что, используя больше параллельных сред, мы позволяем нашему агенту испытывать гораздо больше ситуаций, чем в одной среде) [3].

В работе были рассмотрены следующие библиотеки для обучения агентов:

1. KerasRL
2. Tensorforce
3. OpenAI Baselines
4. Stable Baselines
5. TF Agents

Из всех известных на данный момент библиотек для обучения с подкреплением лучше всего применять в приложении такие библиотеки как Stable Baselines или TF Agents. Данные библиотеки получили наивысшие оценки по параметрам, которые очень сильно влияют на скорость и качество разработки.

Модельное обучение

Для решения поставленной задачи применяется модельное обучение. В модельном обучении с подкреплением используется опыт для построения внутренней модели переходов и непосредственных результатов в среде [3]. Действия затем выбираются путем поиска или планирования в этой модели мира.

С другой стороны, безмодельная обучение с подкреплением использует опыт для непосредственного изучения одной или двух более простых величин (значений состояния / действия или политики), которые могут достичь того же оптимального поведения, но без оценки или использования модели

мира. При наличии политики состояние имеет значение, определенное в терминах будущей полезности, которая, как ожидается, будет накапливаться, начиная с этого состояния. Методы без моделей статистически менее эффективны, чем методы, основанные на моделях, поскольку информация из среды объединяется с предыдущими и, возможно, ошибочными оценками или представлениями о значениях состояний, а не используется напрямую [3].

Обучение на основе моделей пытается смоделировать среду, а затем выбрать оптимальную политику на основе ее изученной модели. В обучении без моделей агент использует метод проб и ошибок для настройки оптимальной политики.

Два основных подхода к представлению агентов с использованием обучения с подкреплением без модели — это оптимизация политики и Q-обучение.

Методы оптимизации или итерации политики. В методах оптимизации политики агент непосредственно изучает функцию политики, которая отображает состояние в действие. Политика определяется без использования функции значения.

Заключение

При рассмотрении методов обучения с подкреплением можно сделать вывод что для разработки приложения лучше всего использовать методы без модели. Это объясняется тем, что в системе с многими агентами агенты не будут иметь возможности построить модель среды. При использовании множества агентов каждый из них будет иметь представление лишь о части среды. Среди методов обучения с подкреплением без модели лучше использовать такие методы как PPO, DQN или гибридные методы, которые включают в себя преимущества как Q-обучения тат и оптимизации политики.

Для эффективной работы модели поведения нарушителя необходимо использовать как можно больше электронно-вычислительных машин. Так как при моделировании скорость возрастает с ростом сложности объекта, который нужно защитить, количества действий, которые могут совершать агенты.

В заключении можно сказать, что грамотная реализация модели среды и поведения нарушителя может обезопасить многие предприятия от угроз проникновения и тем самым обезопасить их.

Литература

1. EAQR: A Multiagent Q-Learning Algorithm for Coordination of Multiple Agents [Электронный ресурс]. – Режим доступа: https://www.hindawi.com/journals/complexity/2018/7172614. – Дата доступа: 21.10.2020
2. Reinforcement learning [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/what-is-reinforcement-learning. – Дата доступа: 25.10.2020
3. Top 7 Python Libraries for Reinforcement Learning [Электронный ресурс]. – Режим доступа: https://analyticsindiamag.com/python-libraries-reinforcement-learning-dqn-rl-ai. – Дата доступа: 28.10.2020
4. Introduction to Various Reinforcement Learning Algorithms. Part 1 (Q-Learning, SARSA, DQN, DDPG) [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287. – Дата доступа: 30.10.2020