М. В. Биза
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **Е. И. Сукач**, канд. техн. наук, доцент

ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ СРЕДЫ CARTPOLE

Машинное обучение представляет собой область компьютерной науки, в которой машины учатся решать задачи, для которых они не были запрограммированы непосредственно [1].

Обучение с подкреплением (RL) – область машинного обучения, в которой обучение происходит через взаимодействия с окружающей средой [2]. Область обучения с подкреплением стала важна и как одна из ведущих областей исследования в сфере искусственного интеллекта, и как инструмент, который находит множество вариантов практического применения. В ней появляется множество алгоритмов, позволяющих эффективно обучать агентов. RL сыграло решающую роль в быстром развитии технологий искусственного интеллекта.

Пусть нужно обучить маятник, расположенный на подвижной платформе, удерживаться в перевернутом состоянии (среда CartPole).

Такие известные методы обучения, как Q-learning и Deep Q-learning, изучают функцию значения, которая показывает ожидаемую сумму вознаграждений, заданных состоянием и действием обучаемого агента.

Q-Learning – это методика обучения с подкреплением без использования моделей. Она обычно считается «самым простым» алгоритмом обучения с подкреплением. Q-Learning использует ранее изученные «состояния», которые были исследованы для рассмотрения будущих ходов, и сохраняет эту информацию в «Q-таблице». Для каждого действия, предпринятого из состояния, Q-table должна включать положительное или отрицательное вознаграждение. Эта форма обучения отлично подходит, когда количество ходов ограничено или среда не сложная, поскольку агент запоминает прошлые ходы и с легкостью повторяет их. Однако для более сложных сред со значительно большим количеством состояний Q-table будет быстро заполняться, что приведет к увеличению времени обучения.

Использование Deep Q-learning предполагает наличие некоторого хранилища с определенным размером, где хранятся последние N опытов агента. При обучении используется случайная выборка определенного размера из памяти воспроизведения, и применяется обновление Q-learning. После воспроизведения опыта агент выбирает и выполняет действие в соответствии с ε-жадной политикой.

Преимуществами данного алгоритма является то, что каждый шаг опыта потенциально используется во многих обновлениях весов нейронной сети, что позволяет повысить эффективность данных; использование случайных выборок нарушает корреляции между выборками и, следовательно, уменьшает дисперсию в обновлениях.

Данные методы имеют много преимуществ и хорошо обучают агентов. Однако можно выделить несколько значительных проблем: они могут иметь большие колебания во время обучения; возникают трудности при большом наборе возможных действий; нужно реализовывать компромисс между разведкой и эксплуатацией.

Все эти проблемы можно решить при помощи градиентов политики. Градиент политики (PG) – это подход к RL, оптимизирующий параметризованную модель политики для ожидаемой отдачи с использованием градиентного подъема. Преимущества градиентов политики: при обучении агента мы просто следуем градиенту для нахождения лучших параметров; параметры настраиваются напрямую; могут изучать стохастические политики.

Недостатком градиентов политики является то, что большую часть времени они сходятся на локальном максимуме, а не на глобальном оптимуме, и их обучение может занять много времени. Но и эти проблемы можно решить, правильно подобрав политику.

При использовании градиентов политики Монте-Карло в обучении предпринимаются следующие действия: вычисление логарифмической вероятности, полученной с помощью функции политики; умножение ее на функцию оценки; обновление веса. Проблема данной политики заключается в усреднении всех действий. Даже если некоторые из них были очень плохими, а балл в итоге получился высоким, данные действия оцениваются как хорошие. Из этого следует, что для получения правильной политики нужно произвести много экспериментов. Это приводит к медленному обучению.

После обучения среды CartPole с помощью описанных ранее алгоритмов можно сделать вывод, что все они требуют больших временных затрат. Однако, если сравнивать градиенты политики Монте-Карло с двумя другими алгоритмами, то, как видно при сравнении рисунка 1, рисунка 2 и рисунка 3, обучение происходит более стабильно.

К тому же, вычисления алгоритмом градиентов политики происходят гораздо быстрее благодаря тому, что не нужно оценивать максимум на каждом шаге.

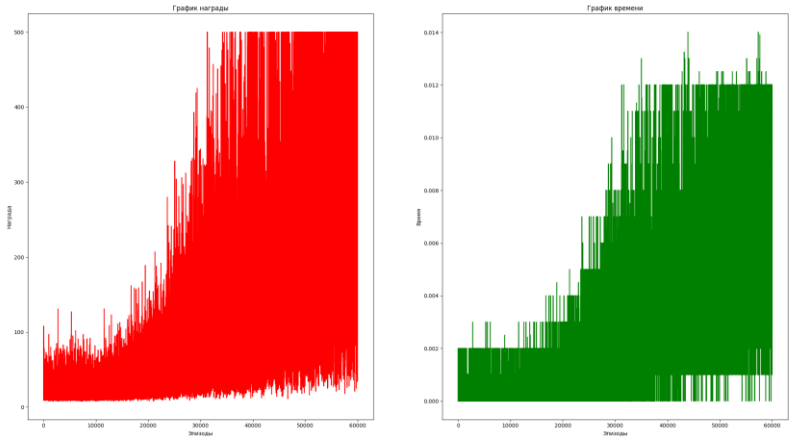

Рисунок 1 – Графики награды и времени для среды CartPole
(с использованием алгоритма Q-learning)

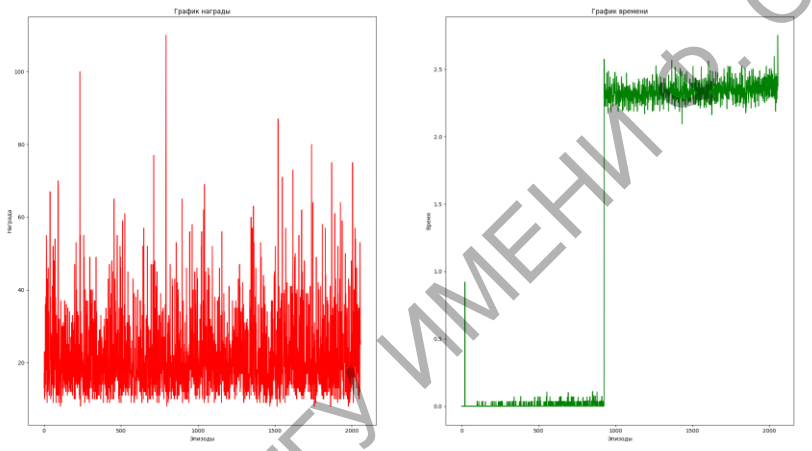

Рисунок 2 – Графики награды и времени для среды CartPole
(с использованием алгоритма Deep Q-learning)

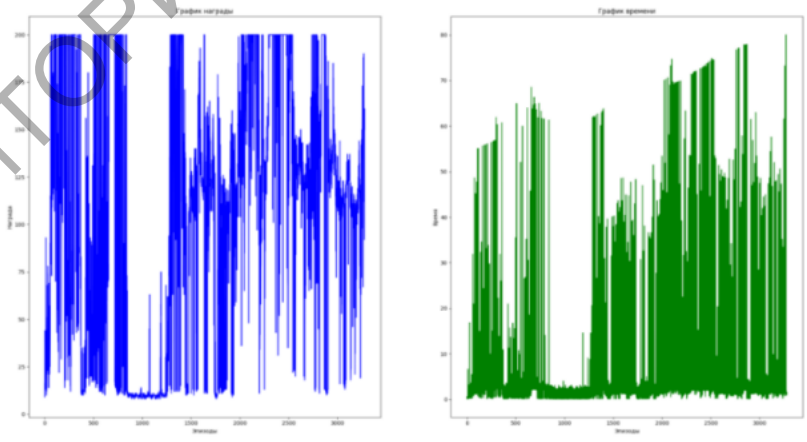

Рисунок 3 – Графики награды и времени для среды CartPole
(с использованием градиентов политики Монте-Карло)

Литература

1. Траск, Э. Грокаем глубокое обучение / Э. Траск. – СПб. : Питер, 2019. – 352 с.

2. Равичандиран, С. Глубокое обучение с подкреплением на Python. OpenAI Gym и TensorFlow для профи / С. Равичандиран. – СПб. : Питер, 2019 – 251 с.

П. С. Бискуб
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **Е. М. Березовская**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ «ФУТБОЛ БЕЛАРУСИ»

Разработка web-приложений в настоящее время идет в стремительном темпе и охватывает различные области человеческой деятельности. Предлагаемая работа посвящена созданию web-приложения «Футбол Беларуси». Приложение реализует следующий функционал:

1. Регистрация и авторизация пользователей (рисунок 1).
2. Просмотр рейтинга команд и турнирной таблицы.
3. Страницу просмотра каждой отдельной команды, которая будет содержать: информацию о команде, список участников команды и информацию о каждом игроке.
4. Добавление игроков, изменения и удаление команд.

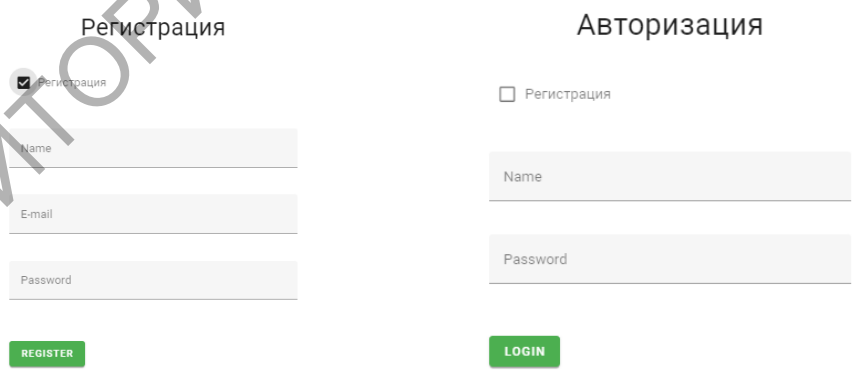

Рисунок 1 – Форма регистрации и авторизации пользователя

В результате работы над проектом разработана и реализована клиентская и серверная часть приложения «Футбол Беларуси». Сервер был разработан с помощью фреймворка Express и базы данных MongoDB.