|  |  |  |
| --- | --- | --- |
| Федеральное государственное автономное образовательное учреждение  высшего образования «Национальный исследовательский  Нижегородский государственный университет им. Н.И. Лобачевского» НИИМ Нижегородского университета | | |
|  | | **УТВЕРЖДАЮ**  Профессор кафедры  ИАНИ ННГУ, д.т.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Старостин  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |
| Научно-технический отчет  **РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**  **«Разработка агента в OpenAI Gym с использованием**  **методик обучения с подкреплением»** | | |
|  |  | |
|  |  | |
|  |  | |

Нижний Новгород

2021

СОДЕРЖАНИЕ

[Введение 3](#_Toc73115554)

[1. Содержательная постановка задачи 4](#_Toc73115555)

[1.1. Описание среды 4](#_Toc73115556)

[2. Входные данные 5](#_Toc73115557)

[3. Выходные данные 6](#_Toc73115558)

[3.1. Запуск обучения моделей 6](#_Toc73115559)

[3.2. Запуск обученной модели 6](#_Toc73115560)

[4. Разработка моделей обучения 8](#_Toc73115561)

[4.1. Собственные модель обучения 8](#_Toc73115562)

[4.2. Q-Learning модель обучения 8](#_Toc73115563)

[4.3. Нейросетевая модель обучения 8](#_Toc73115564)

[5. Процесс обучения агента 10](#_Toc73115565)

[5.1. Системные сообщения 10](#_Toc73115566)

[5.2. Возможные ошибки во время обучения агента 11](#_Toc73115567)

[Заключение 12](#_Toc73115568)

[Ссылки на источники 12](#_Toc73115569)

# Введение

В рамках данной исследовательской работе разрабатывается агент в OpenAI Gym с использованием методик обучения с подкреплением.

В рамках проекта рассмотрено несколько подходов обучения агента для выполнения его задачи в среде LunarLander-v2, таких как: собственная модель обучения, Q-Learning модель обучения и нейросетевая модель обучения.

# Содержательная постановка задачи

Задачей данного проекта является изучение и реализация алгоритмов, которые будут осуществлять обучение посадке модуля в среде LunarLander-v2, при этом максимизируя количество очков награды.

## Описание среды

Среда представляет собой 2D приложение с летающим модулем и поверхностью с выделенной площадкой для посадки. Модуль необходимо посадить на площадку, избегая его крушения. В начале каждого эпизода модулю сообщается случайный импульс. На модуль можно воздействовать путем включения одного из двигателей за такт. Очки награды начисляются за успешное приземление модуля на поверхность. Успешным считается мягкое приземление на обе ножки модуля на помеченную флажками площадку. Вычитаются очки за каждое использование двигателей (по -0.03 за боковые и -0.3 за главный) и за жёсткое приземление (приземление с большой скоростью и/или крушение).

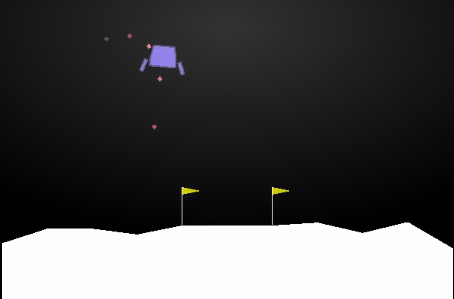


Рис 1. Внешний вид среды LunarLander-v2

# Входные данные

Входными данными является сама среда LunarLander-v2 и механизмы взаимодействия с ней через код. Среда реализована на языке Python, и её полный код реализации находится в открытом доступе [[1](#_Ссылки_на_источники)]. Описание основного функционала среды на русском языке находится в отчёте-аннотации.

# Выходные данные

Выходными данными работы являются графики обучения агента разными моделями, видеозаписи фрагментов обучения агента, а также веса нейросети, полученные в ходе обучения.

## Запуск обучения моделей

Для того чтобы запустить обучение агента на ОС Windows, необходимо:

* открыть командную строку cmd и перейти в папку …/LunarLanderProject/Code командой cd или же зайти в эту папку через проводник и открытый через PowerShell нажатием Shift+правая кнопка мыши;
* далее ввести команду «python learn\_models.py», выбрать необходимую модель и указать число запусков.

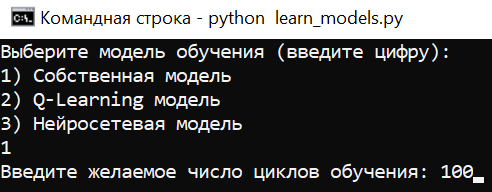


Рис 2. Пример запуска обучения агента через cmd

Также можно запустить данный файл, открыв его через среду разработки, например, Spyder.

## Запуск обученной модели

Для того чтобы запустить демонстрацию работы обученного агента на ОС Windows, необходимо:

* открыть командную строку cmd и перейти в папку …/LunarLanderProject/Code командой cd или же зайти в эту папку через проводник и открытый через PowerShell нажатием Shift+правая кнопка мыши;
* далее ввести команду «python lunar\_lander.py» и указать число запусков.
* указать, сохранять ли диаграммы по результатам запусков.

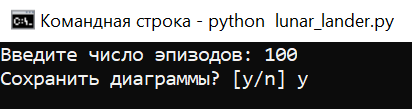


Рис 3. Пример запуска демонстрации обученного агента через cmd

Также можно запустить данный файл, открыв его через среду разработки, например, Spyder.

В проекте собран запускаемый файл lunar\_lander.exe для того, чтобы любой пользователь мог посмотреть результат обучения модели без установки Python на своё устройство. В папке «Trained Lander» необходимо найти файл lunar\_lander.exe и создать его ярлык, как в примере ниже.

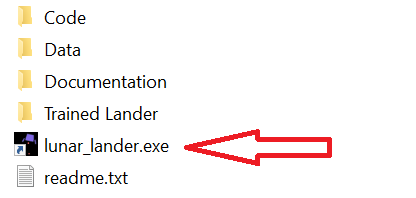


Рис 4. Пример создания ярлыка приложения

При запуске пользователю нужно лишь указать желаемое количество эпизодов игры и возможность сохранения графиков результата. При этом результаты запусков будут выводиться в консоль.



Рис 5. Результат запуска приложения

# Разработка моделей обучения

## Собственные модель обучения

В качестве самобытных подходов к обучению были рассмотрены табличная модель и модель, основанная на RDF (Random Decision Forest).

Табличная модель представляет собой модель обучения, основанную на полном переборе таблицы состояний приземляемого модуля, действий над ним и оценки его действий. Выбираются лучшие действия, в соответствие с наградой за траекторию, из строк таблицы на основании евклидового расстояния между состояниями.

Во второй модели схожая логика, только поиск действий происходит с помощью RDF, который построен на основе той же таблицы.

## Q-Learning модель обучения

Данная модель обучения представляет собой один из классических подходов к обучению с подкреплением (Reinforcement Learning), применяемый при агентном подходе.

На основе получаемого от среды вознаграждения и функцию расчёта полезности Q формируется таблица, из которой по текущему состоянию модуля можно выбрать наиболее выгодное действие с точки зрения функции полезности. По мере обучения добавляются новые состояния и корректируются значения в таблице, что впоследствии дает возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия модуля со средой.

## Нейросетевая модель обучения

Данная модель обучения состоит из двух подсетей: «Актера» и «Критика». Модель «Актер» выполняет задачу изучения того, какие действия следует совершить в конкретном наблюдаемом состоянии окружающей среды. После завершения этих действий и получения очков награды, решения «Актера» оцениваются «Критиком». Таким образом, данные подсети дополняют друг друга, основываясь на вердиктах.

Идея такого подхода взята из статьи об обучении агента в среде LunarLander-v2 [[2](#_Ссылки_на_источники)].

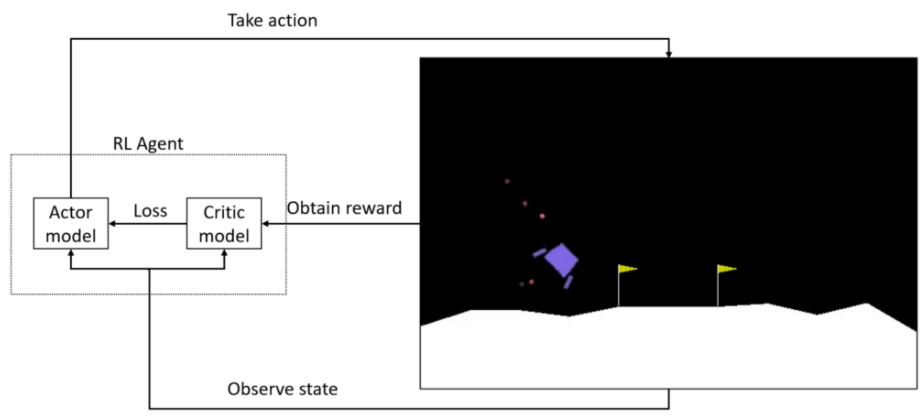


Рис 6. Структура модели «Актёр – Критик»

# Процесс обучения агента

## Системные сообщения

В случае успешного запуска агента, откроется окно с рендером среды LunarLander.

При запуске обучения агента из среды разработки справа в окне консоли будет вестись вывод результатов эпизодов: текущий номер эпизода, награда и среднее. При запуске через cmd или PowerShell аналогичный вывод будет осуществлять прямо в этом же консоли. При запуске обучения собственной моделью дополнительно будут выводиться сообщения о добавлении траекторий и переобучении дерева.

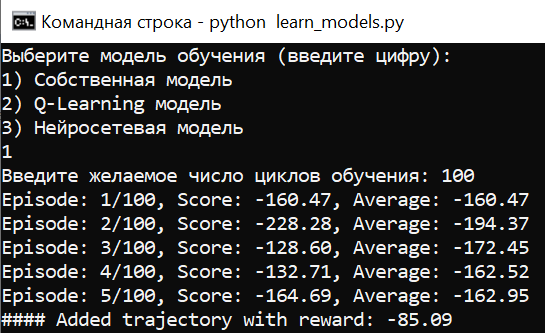


Рис 7. Пример вывода в консоль при запуске обучения собственной моделью

При запуске демонстрации работы обученного агента вывод осуществляется по той же аналогии, только немного видоизменён.

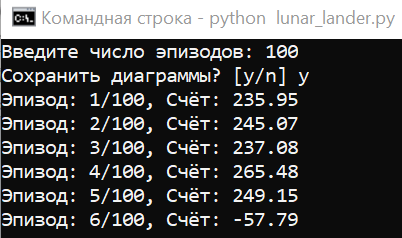


Рис 8. Пример вывода в консоль при запуске обученного агента

## Возможные ошибки во время обучения агента

В процессе эксплуатации агента аварийные ситуации, выражающиеся в отсутствии отклика, могут возникать по следующим причинам:

* Потеря питания ПК;
* Принудительное закрытие приложения во время обучения агента;
* Работа на ПК, характеристики которого не удовлетворяют системным требованиям.

При потере питания, необходимо заново запустить ПК и перезапустить среду разработки. В этом случае данные обучения могут быть утеряны.

В случае работе на ПК, характеристики которого не удовлетворяют системным требованиям, возможно подвисание окна, в котором происходит визуализация среды, а также возможен «вылет» из программы, с дальнейшей потерей данных.

При желании пользователя досрочно завершить обучение – необходимо перезапустить ядро Python. Для этого необходимо на верхней панели найти вкладку «Consoles», открыть ее и нажать «Interrupt kernel».

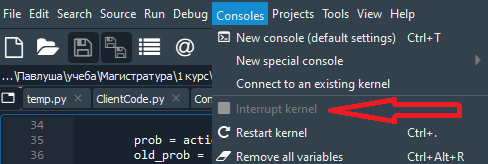


Рис 9. Иллюстрация перезагрузки ядра в среде разработки Spyder

Далее необходимо нажать «Restart kernel» (находится под Interrupt kernel). Ядро перезапустится, однако данные обучения будут утеряны.

Чтобы досрочно завершить работу при запуске через консоль, необходимо нажать Ctrl+C.

# Заключение

Была поставлена задача обучения агента в среде LunarLander-v2 при помощи нескольких моделей обучения с целью выполнения его задачи. Агент получил возможность обучаться при помощи одной из выбранных моделей.

В ходе работы было установлены следующие факты:

* Собственные модели обучения оказались недостаточно эффективны по ряду признаков: по памяти в работе RDF, по времени поиска по таблице, а также по невозможности эффективной оценки действий такими методами.
* Q-Learning модель обучения оказалась так же малоэффективной. По поведению агента можно сделать вывод, что он обучается, но средний показатель награды в процессе обучения не возрастает. Скорее всего это связано с тем, что классический Q-Learning алгоритм не подходит для нашего агента, так как у него много параметров.
* Нейросетевой подход к обучению оказался эффективным, и в ходе обучения была заметна тенденция увеличения среднего значения награды. Однако нейросетевой подход потребовал очень долгого времени работы.

По итогу обучения получены веса нейросети, которые были использованы для демонстрации хорошей работы агента через созданное приложение.

# Ссылки на источники

1. Код среды LunarLander-v2 на GitHub: <https://github.com/openai/gym/blob/master/gym/envs/box2d/lunar_lander.py>
2. Обучение агента на основе нейросети «Актер - Критик»: <https://pylessons.com/LunarLander-v2-PPO/>
3. Репозиторий проекта: <https://github.com/erofale/SeminarProject>
4. Архив с приложением: <https://disk.yandex.ru/d/i4AgqtpjBTCq_g>