

Обучение с подкреплением в графическом трехмерном окружении

Федорова Анна, БПМ151
научный руководитель: Булычев Д.Ю.
научный консультант: Пилюгин К.С.

Высшая школа экономики

2019 г.

Обучение с подкреплением



OpenAI Gym¹

- Нет интерфейса для создания новых окружений.
- Не содержит готовые реализации алгоритмов обучения.

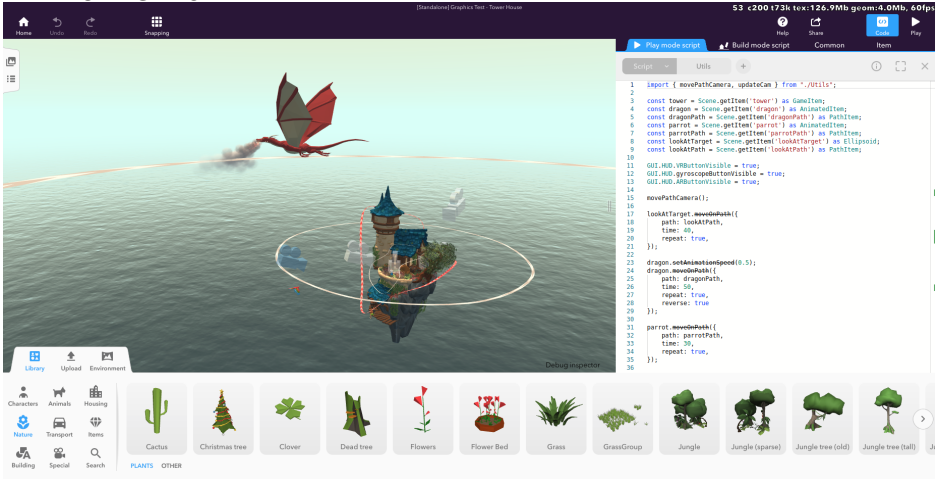
Unity ML Agents²

- Реализован один алгоритм обучения.
- Сложный интерфейс.

¹G. Brockman et al., “OpenAI Gym,” In: arXiv preprint ArXiv160601540 Cs, Jun. 2016

²A. Juliani et al., “Unity: A General Platform for Intelligent Agents,” In: arXiv preprint ArXiv180902627 Cs Stat, Sep. 2018

Платформа для создания интерактивного 3D контента.



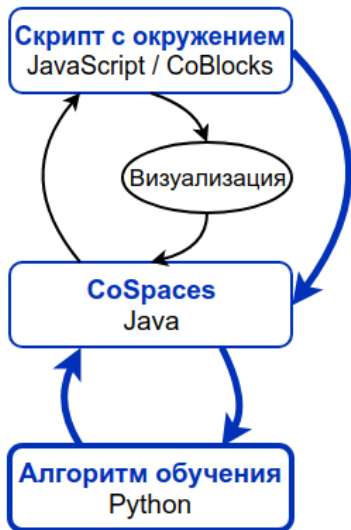
Интегрировать в приложение *CoSpaces* модуль для создания и решения задач обучения с подкреплением.

- Разработать архитектуру нового модуля в рамках приложения.
- Разработать программный интерфейс для создания обучаемых агентов.
- Создать модуль обучения и библиотеку базовых алгоритмов.
- Реализовать запуск обученной модели из веб-версии приложения.
- Протестировать возможности обучения.



Возможности приложения CoSpaces:

- Создание сцен из 3D объектов с использованием физического симулятора.
- Описание поведения и взаимодействия объектов в скрипте.
- Исполнение скрипта в режиме проигрывания с графической визуализацией каждого такта.



- Разработать скриптовый программный интерфейс для идентификации агента.
- Реализовать новый режим исполнения скрипта без отрисовки.
- На языке *Python* создать модуль обучения с подкреплением.
- Разработать коммуникационный протокол между приложением и модулем обучения.

```
let cart = Scene.getItem("cart") as Capsule;
let pole = Scene.getItem("pole") as Capsule;
function reset(): number[] {
  cart.transform.set(new Transform());
  pole.transform.set(new Transform());
  return getState();
}
function step(action: number[]): void {
  cart.physics.velocity = new Vector3(action[0], 0, 0);
}
function isFinal(): boolean {
  return (abs(cart.transform.position.x) > pThreshold) ||
    (angle(pole.transform.axisZ) > aThreshold);
}
function response(): number[] {
  let done = isFinal() ? 1 : -1;
  let reward = done == 1 ? -10 : 1;
  return getState().concat([reward * 10, done]);
}
ML.createLearningEnvironment({
  stateSpaceSize: 4,
  actionSpaceSize: 1,
  action: action => step(action),
  response: () => response(),
  reset: () => reset()
});
```

Для запуска обучения:

- В скрипте вызвать функцию обучения согласно предложенному программному интерфейсу.
- Выбрать и настроить алгоритм обучения.
- Запустить скрипт в одном из предложенных режимов.

Режимы исполнения скрипта

Для оптимизации времени обучения реализован режим исполнения скрипта без отрисовки.

Поддерживается свободное переключение между двумя режимами исполнения без потери прогресса обучения.

Режим исполнения	Одна итерация
Режим с отрисовкой	$5 \cdot 10^{-2} \text{ с.}$
Режим без отрисовки	$3.5 \cdot 10^{-4} \text{ с.}$

Реализованы базовые алгоритмы на языке *Python* на основе библиотеки *Tensorflow*:

- DQN³,
- DDPG⁴,
- PPO⁵.

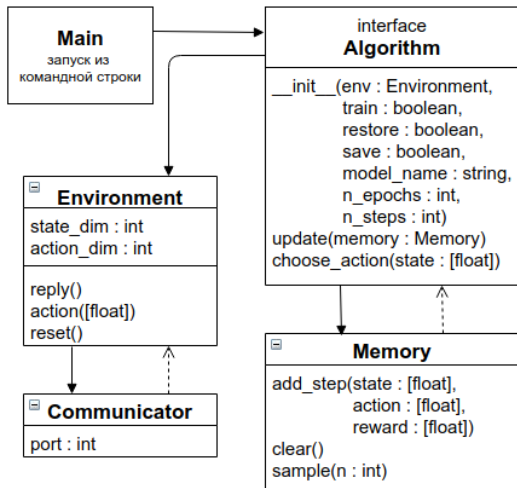
Перед запуском скрипта есть возможность настроить дополнительные параметры, такие как:
выбор обучающего алгоритма, имя модели,
восстановить/сохранить модель, количество эпох, максимальная длина эпизода и др.

⁴V. Mnih et al., “Playing Atari with Deep Reinforcement Learning”

⁵T. P. Lillicrap et al., “Continuous control with deep reinforcement learning”

⁶J. Schulman et al., “Proximal Policy Optimization Algorithms”

Алгоритмы обучения с подкреплением



Обучение
происходит локально.

Для запуска нового
алгоритма необходимо
написать алгоритм в
классе-наследнике
интерфейса *Algorithm* и
указать к нему путь.

Коммуникационный протокол

Коммуникация Java и Python: клиент-серверное взаимодействие через сокет.

Модель передаваемых данных:



Обученная модель может быть конвертирована в формат JSON и загружена на удаленный сервер.

Для запуска модели в веб-приложении используется библиотека Tensorflowjs⁶ для JavaScript.

При запуске скрипта происходит подключение библиотеки, запуск модели и последующее моделирование поведения агента.

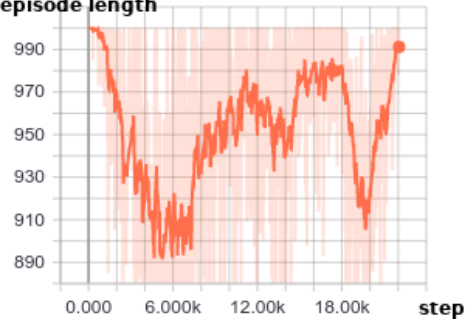
⁶"TensorFlow.js: Machine Learning for the Web and Beyond", ArXiv Preprint ArXiv:1901.05350, Feb. 2019.

PushBlock
Half-Cheetah

BalanceBall
CartPole

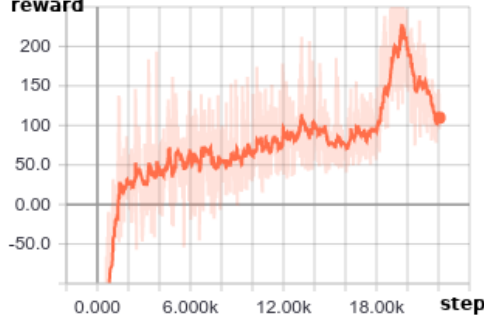
Визуализация статистики

episode length



Изменение длины
эпизода от количества
эпизодов

reward



Изменение суммарной
награды от количества
эпизодов

Результаты

Окружение	Количество эпох	Время обучения	Длина эпизода	Награда	OpenAI Gym
<i>CartPole (DQN)</i>	10	3 мин.	1000	1000 1000 шагов	3 мин.
<i>BalanceBall (DDPG)</i>	50-70	20 мин.	1000	497.3 1000 шагов	
<i>Half-Cheetah (PPO)</i>	10k-15k	20 ч.	1000	253.7 631 шаг	12 ч.
<i>PushBlock (PPO)</i>	> 25k	35 ч.	2000	299.4 825 шагов	

В результате проделанной работы:

- В приложении CoSpaces создан модуль обучения с подкреплением,
- Разработан программный интерфейс для создания обучаемых агентов,
- Реализованы 3 базовых алгоритма,
- Обученная модель может быть запущена из веб-версии приложения.

Среда отличается простым интерфейсом и графической визуализацией процесса обучения.

Настраиваемые параметры обучения:

- Количество эпох обучения,
- Максимальная длина эпизода,
- Режим обучения или воспроизведения,
- Количество скрытых слоев в нейронных сетях,
- Порт для сокета,
- Путь к скрипту с алгоритмом,
- Имя модели,
- Восстановить модель,
- Сохранить модель.