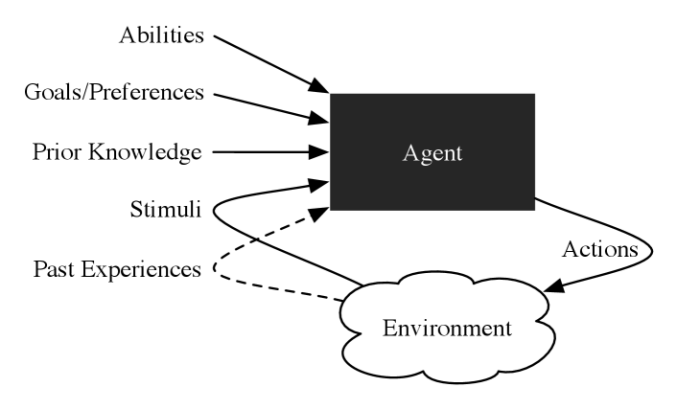
Применение техник обучения с подкреплением по учебной программе и самостоятельной игры для конкурентных сред в казахских национальных играх

May 11, 2024

Динара Жусупова
Руководитель:Елена Кантонистова
Машинное обучение и высоконагруженные системы, НИУ Высшая школа экономики

Агент. Среда



Алгоритмы обучения с подкреплением могут обучать агентов, которые решают проблемы в сложных, интересных средах.

RL

RL – это эффективный инструмент, который помогает системам искусственного интеллекта (ИИ) достигать оптимальных результатов в полно/частично наблюдаемых средах.

Агент и среда

Агент взаимодействует со средой, которая задаётся зависящим от времени состоянием. Агенту в каждый момент времени в общем случае доступно только некоторое наблюдение текущего состояния среды. Сам агент задаёт процедуру выбора действия по доступным наблюдениям; эта процедура называется стратегией или политикой. Процесс взаимодействия агента и среды задаётся динамикой среды, определяющей правила смены состояний среды во времени и генерации награды.

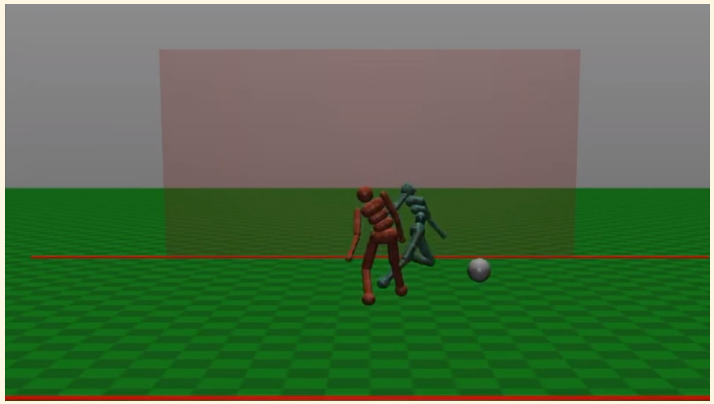
Self-play (самостоятельная игра)

В конкурентной многоагентной среде агенты обучаются с помощью техники self-play.

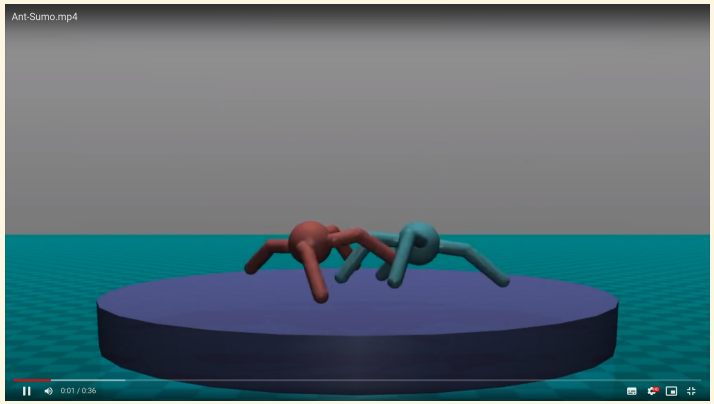
Обучение агентов в мультиагентных средах

T. Bansal, UM. Amherst, J. Pachocki, S. Sidor, I. Sutskever, I. Mordatch. EMERGENT COMPLEXITY VIA MULTI-AGENT COMPETITION https://arxiv.org/abs/1710.03748

В этой статье представлены несколько конкурентных мультиагентных сред, в которых агенты соревнуются в трехмерном мире с симулированной физикой. Обученные агенты изучают широкий спектр сложных и интересных навыков, хотя сама среда относительно проста. Авторы рассматривают два трехмерных тела агента: муравья и гуманоида. Муравей представляет собой четвероногое тело с 12 степенями свободы и 8 приводящими в движение суставами. Гуманоид имеет 23 степени свободы и 17 приводимых в действие суставов.



Игра «Нападай и защищайся» для гуманоидов

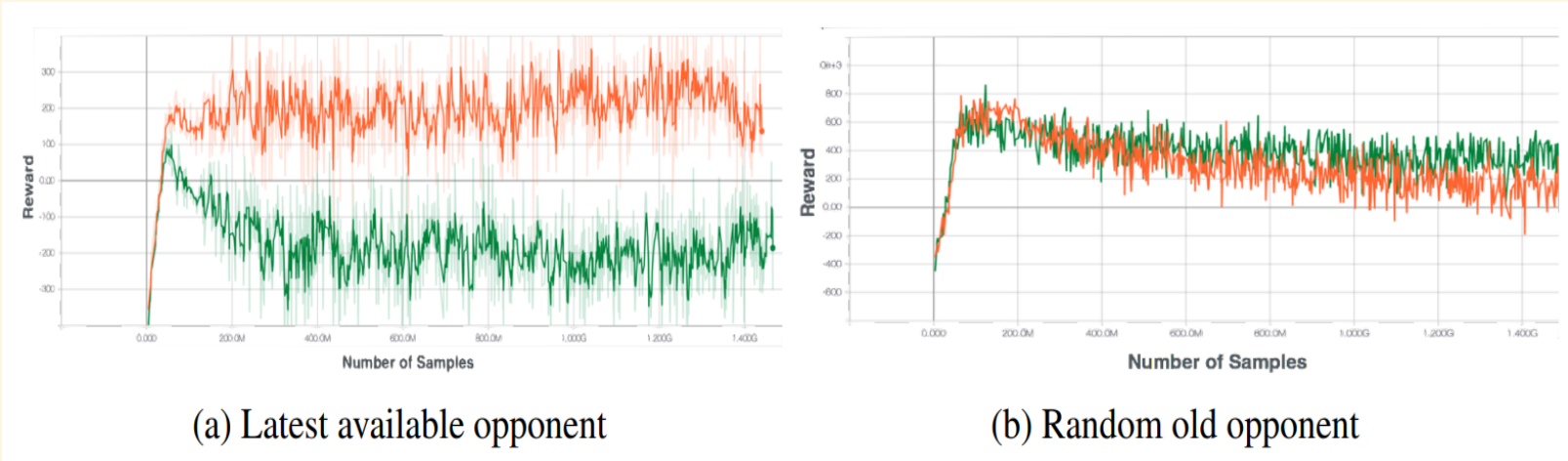


Игра сумо для муравьев

Обучение конкурентных агентов. Сэмплирование оппонентов

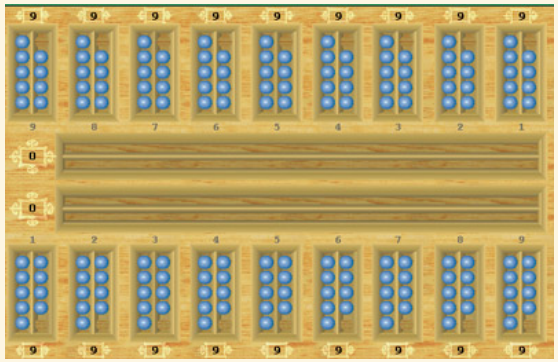
Навыки противников, с которыми сталкиваются во время обучения, могут оказать существенное влияние на обучение агентов.

Обучение агентов против самого последнего противника приводит к дисбалансу в обучении, когда один агент становится более опытным, чем другой агент, на ранних этапах обучения, а другой агент не может восстановиться.



(a) Latest available opponent (b) Random old opponent

Об игре «Тогыз кумалак» (Тоғызқұмалақ)



Описание игры

Казахская национальная логическая настольная игра на доске, в которой 18 игровых и две накопительные лунки. В 2020 году ЮНЕСКО включило эту игру в репрезентативный список нематериального наследия. Доска для игры в тогыз кумалак имеет по 9 игровых лунок для каждого игрока. Кроме этого, в середине доски располагаются две большие лунки для сбора выигранных камней. Количество камней - 162 штуки. Для обозначения лунки-туздыка используются специальные знаки, при их отсутствии могут использоваться два камня отличающихся по форме или цвету от игровых камней.

- **Количество игроков:** 2
- **Исходная позиция:** Перед началом игры в каждую игровую лунку раскладывают по 9 камней. Накопительная лунка пуста. Каждому игроку принадлежит ближний к нему ряд из 9-ти лунок (уй – дом) и одна накопительная лунка (казан – котёл), располагающаяся ближе к игроку или по правую руку.

Основные правила игры

Ходы

Ходы делают по очереди. Право первого хода взаимно оговаривается или разыгрывается жребием, начинающий игру садится с белой стороны. Во время своего хода игрок берёт все камни из любой своей лунки «дом» и, начиная с этой же лунки, раскладывает их по одному против часовой стрелки в свои и чужие дома. Если в исходной лунке только один камень, то он перекладывается в следующую лунку.

Выигрыш камней

Если последний кумалак попадает в дом соперника и количество кумалаков в нём становится чётным, то кумалаки из этого дома переходят в казан игрока, совершившего ход.

Туздык

Туздык — выигранная лунка на стороне соперника. Если при ходе игрока А последний кумалак попадает в дом игрока Б и в нём после этого оказывается три кумалака, то этот дом объявляется туздыком игрока А (каз. туздык уй). Эти три кумалака попадают в казан игрока А. В последующем каждый кумалак, попавший в туздык, переходит в казан игрока А. Существует 3 основных правила взятия туздыка:

1. игрок не может завести себе туздык в самом последнем (девятом) доме соперника,
2. игрок не может завести себе туздык в доме с таким же порядковым номером, который имеет лунка-туздык соперника,
3. каждый игрок в течение игры может завести себе только один туздык.

Отказаться от туздыка или изменить его положение нельзя, он действует до конца игры.

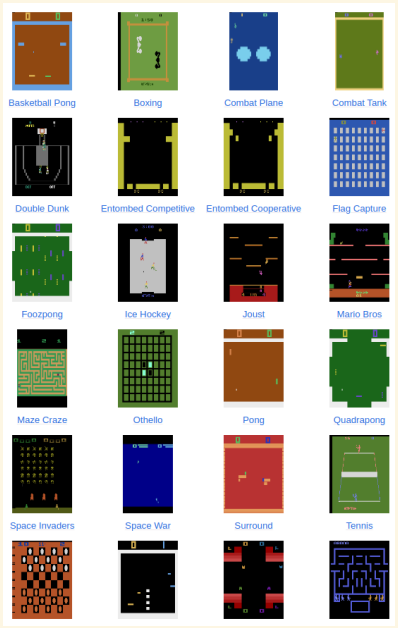
Стек

- **Для создания мультиагентной среды:** PettingZoo https://pettingzoo.farama.org/
- **Для обучения конкурентных агентов:** Tianshou https://tianshou.org/en/stable/index.html
- **Для создания веб приложения игры:** Dash https://dash.plotly.com/

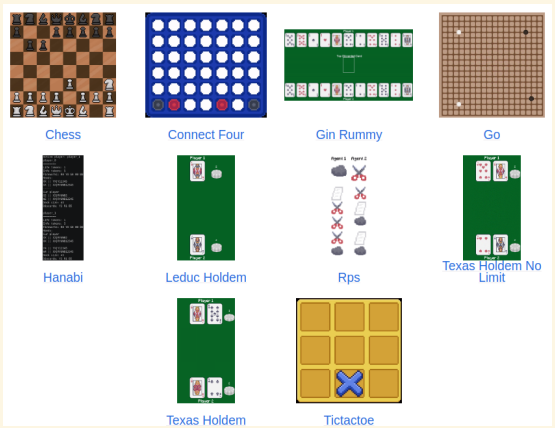
PettingZoo

PettingZoo

PettingZoo - это простой питонический интерфейс, способный отображать общие проблемы многоагентного обучения с подкреплением (MARL - multi-agent reinforcement learning). PettingZoo включает в себя широкий спектр эталонных сред, полезных утилит и инструментов для создания собственных пользовательских сред.

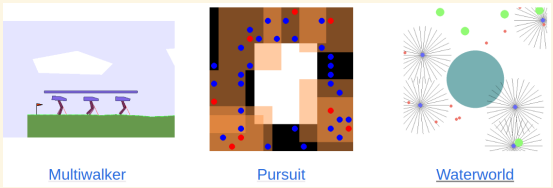


Atari



Classic

Butterfly



SISL

PettingZoo

PettingZoo имеет хорошие туториалы https://pettingzoo.farama.org/tutorials/, подробную документацию https://pettingzoo.farama.org/content/basic_usage/.

- **AEC API** По умолчанию PettingZoo моделирует игры как среды Agent Environment Cycle (AEC). Это позволяет PettingZoo представлять любой тип многоагентной игры, который может быть рассмотрен RL.
- **Parallel API** В дополнение к основному API есть дополнительный Parallel API для сред, где все агенты выполняют одновременные действия и наблюдения.

Tianshou

Комплексная функциональность

RL Platform	GitHub Stars	# of Alg. (1)	Custom Env	Batch Training	RNN Support	Nested Observation	Backend
Baselines	Stars 15k	9	✔ (gym)	▬ (2)	✔	✘	TF1
Stable-Baselines	Stars 4.1k	11	✔ (gym)	▬ (2)	✔	✘	TF1
Stable-Baselines3	Stars 8k	7 (3)	✔ (gym)	▬ (2)	✘	✔	PyTorch
Ray/RLlib	Stars 31k	16	✔	✔	✔	✔	TF/PyTorch
SpinningUp	Stars 9.7k	6	✔ (gym)	▬ (2)	✘	✘	PyTorch
Dopamine	Stars 10k	7	✘	✘	✘	✘	TF/JAX
ACME	Stars 3.4k	14	✔ (dm_env)	✔	✔	✔	TF/JAX
keras-rl	Stars 5.5k	7	✔ (gym)	✘	✘	✘	Keras
rlpyt	Stars 2.2k	11	✘	✔	✔	✔	PyTorch
ChainerRL	Stars 1.2k	18	✔ (gym)	✔	✔	✘	Chainer
Sample Factory	Stars 745	1 (4)	✔ (gym)	✔	✔	✔	PyTorch
Tianshou	Stars 7.5k	20	✔ (Gymnasium)	✔	✔	✔	PyTorch

Tianshou - это платформа обучения с подкреплением, основанная на чистом PyTorch. В отличие от существующих библиотек обучения с подкреплением, которые в основном основаны на TensorFlow, имеют множество вложенных классов, недружественный API или медленную скорость, Tianshou предоставляет высокоскоростную среду и Pythonic API для создания агента глубокого обучения с подкреплением.

Tianshou

Алгоритмы

Поддерживаемые алгоритмы интерфейса включают в себя:

- **DQNPolicy** Deep Q-Network
- **RainbowPolicy** Rainbow DQN
- **A2CPolicy** Advantage Actor-Critic
- **PPOPolicy** Proximal Policy Optimization
- **DDPGPolicy** Deep Deterministic Policy Gradient
- **TRPOPolicy** Trust Region Policy Optimization

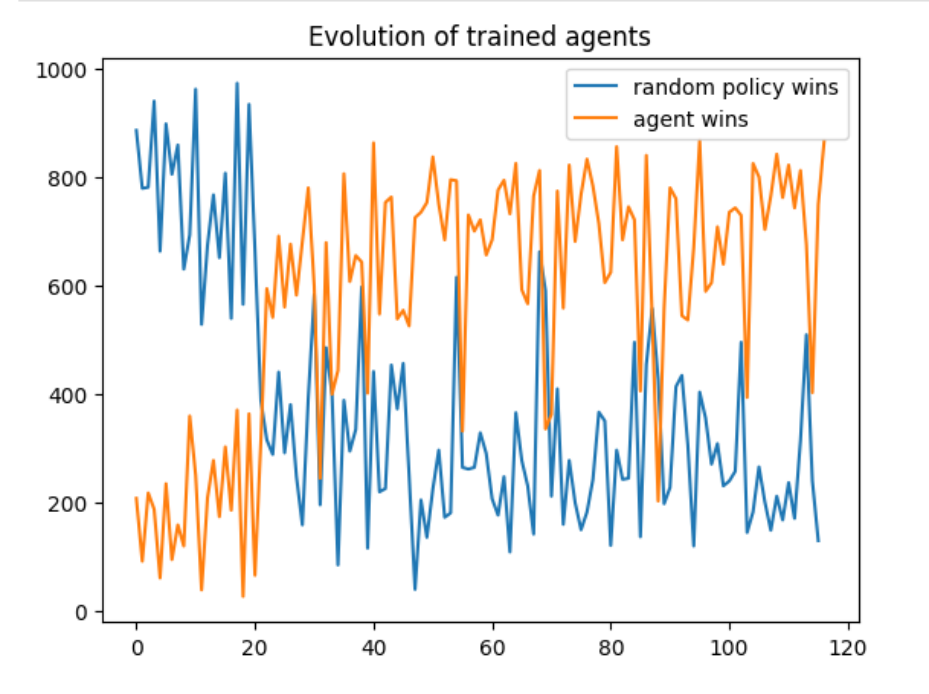
Dash

Dash - это оригинальный фреймворк low-code для быстрого создания приложений с данными на Python.

Dash fundamentals

- **Layout** Приложение Dash состоит из двух частей. Первая часть — это «макет», который описывает, как выглядит приложение.
- **Callbacks** Инструмент для создания приложений Dash с использованием функций callback: функций, которые автоматически вызываются Dash при каждом изменении свойств входного компонента, чтобы обновить какое-либо свойство в другом компоненте (выходном).
- **Interactive Visualizations** Модуль Dash Core Components (dash.dcc) включает компонент Graph под названием dcc.Graph. dcc.Graph визуализирует интерактивные визуализации данных, используя графическую библиотеку JavaScript с открытым исходным кодом plotly.js. Plotly.js поддерживает более 35 типов диаграмм и отображает диаграммы как в формате SVG векторного качества, так и в высокопроизводительном формате WebGL.

Детали обучения



Method

Deep Q Learning (DQN)

batch size = 256
epochs = 150

Net

hidden-sizes = [256, 512, 512, 256]

Сравнение агентов

Лучшая модель №99 побеждает 87.5% противников.

Веб приложение игры

Проект: https://github.com/zhus-dika/togyz-qumalaq-agent.git

Запуск приложения: python app/app.py