

1 Реферат

Перечень ключевых слов: обучение с подкреплением, reinforcement learning, RL, распределение ресурсов в облаке, облачные технологии, облачные ресурсы, Tetris, OpenAI Gym, TensorFlow, KerasRL.

Краткое описание объекта исследования: особенности выделения ресурсов при работе с облачными сервисами.

Краткое описание предмета исследования: применимость обучения с подкреплением для задачи распределения ресурсов в облаке.

Цель проекта: исследование применимости обучения с подкреплением в задачах распределения облачных ресурсов. Сравнение данного подхода с другими методами решения задачи.

Метод или методология проведения работы: метод эксперимента.

Результаты проекта: выяснение границ применимости обучения с подкреплением для решения задач распределения облачных ресурсов.

Апробация результатов:

2 Содержание

Содержание

1	Реферат	1
2	Содержание	2
3	Основные термины, определения и сокращения	3
4	Введение	4
5	Обзор и анализ источников	5
6	Выбор методов, алгоритмов, моделей для решения поставленных задач	6
	Список использованных источников	7
7	Приложения	9

3 Основные термины, определения и сокращения

IT (произносится ай-ти, сокращение от англ. Information Technology) — информационные технологии

RL (англ. reinforcement learning) — обучение с подкреплением

4 Введение

В первой части работы описано применение обучения с подкреплением для визуализации компьютерной игры "Тетрис"[1]. Эта игра представляет собой клетчатое поле шириной 10 клеток и высотой 20 клеток. В верхней части поля друг за другом появляются клетчатые фигурки, состоящие из 4 клеток (тетрамино). Фигурки имеют форму, напоминающую форму букв "I" "Z" "L" "T" а также квадрат из четырех клеток. Пользователь имеет возможность поворачивать фигурку на 90°, а также двигать ее по горизонтали во время падения. В случае заполнения одной из строк частями фигурок строка "исчезает": все фигурки выше нее опускаются на одну строку вниз. Каждая "исчезнувшая" строка приносит игроку 1 очко.

Во второй части работы обучение с подкреплением применено для решения задач распределения облачных ресурсов.

Актуальность

Облачные технологии позволяют обеспечить круглосуточную и бесперебойную работу интернет-сервисов, что делает их востребованными во всех сферах IT-индустрии. Облачными вычислениями занимаются Amazon, Google, Huawei и другие крупнейшие информационные компании[2][3]. В 2020 году мировой рынок облачных вычислений оценивается в 289.25 миллиардов долларов[4]. Распределение облачных ресурсов — одна из важнейших задач облачных вычислений.

Предмет исследования

Возможность использования обучения с подкреплением для решения задачи распределения ресурсов облака.

Методы исследования

Экспериментальное сравнение показателей RL в ходе решения задачи распределения облачных ресурсов с иными используемыми на практике способами. Для наглядности в работе также решена близкая задача: автоматическая игра в "Тетрис" с помощью обучения с подкреплением.

Цели и задачи работы

Определение эффективности обучения с подкреплением в задаче распределения ресурсов в облаке.

Новизна и достоверность полученных результатов

Теоретическая значимость

Практическая ценность

В случае превосходства RL над другими методами в рамках решения задачи распределения облачных ресурсов применение данного способа машинного обучения способно сократить нагрузку на сервера, предоставляющие доступ к облачным сервисам. Это позволит уменьшить расходы компаний на поддержку их работоспособности, а также расходы на производство при сокращении количества серверов. Проект имеет практическую ценность для экологии: уменьшение расходов электроэнергии приведет к уменьшению углеродного следа компаний.

5 Обзор и анализ источников

Первая часть курсовой работы посвящена автоматической игре в "Тетрис" с помощью обучения с подкреплением. Рассмотрим исследования данной задачи и ее решения. В статье "Tetris is Hard, Even to Approximate"[5] доказывается, что игра Тетрис является NP-полной задачей. Это одна из причин схожести данной игры с распределением ресурсов в облаке[6]. В статье Playing the Original Game Boy Tetris Using a Real Coded Genetic Algorithm[7] используется генетический алгоритм для симуляции игры в тетрис. В данной работе метриками успеха автор считает максимальное число удаленных строк до поражения и среднее число удаленных строк у запущенного несколько раз алгоритма. Обе метрики значительно уступают роевым оптимизациям, продемонстрированным в работах Apply ant colony optimization to tetris[8] и Swarm tetris: Applying particle swarm optimization to tetris[9]. Примером использования RL для игры в Тетрис является статья A deep reinforcement learning bot that plays tetris[10].

6 Выбор методов, алгоритмов, моделей для решения поставленных задач

Для демонстрации работы обучения с подкреплением на примере игры "Тетрис" требовалось выбрать среду для симуляции игры, а также библиотеку для реализации машинного обучения.

В качестве среды был рассмотрен симулятор устройства для игр "Game Boy" PyBoy[11]. Однако он был отвергнут в пользу более популярной и более простой в использовании библиотеки gym-tetris[12], являющейся частью OpenAI Gym[13] — среды для симуляции известных компьютерных игр и физических задач.

При выборе библиотеки были рассмотрены pyqlearning[14] и Tensorforce[15]. Однако выбрана была библиотека KerasRL[16], надстройка над фреймворком TensorFlow[17]. Выбор был сделан в пользу KerasRL из-за совместимости со средой OpenAI Gym.

Список использованных источников

- [1] accel-brain, chimera0 [Электронный ресурс] / Pypi. Режим доступа: <https://pypi.org/project/pyqlearning/>, свободный. (дата обращения: 01.02.2022)
- [2] Arif Mohamed. A history of cloud computing [Электронный ресурс]: Computerweekly, 2018 – Режим доступа: <https://www.computerweekly.com/feature/A-history-of-cloud-computing>, свободный. (дата обращения: 11.12.2021)
- [3] Baekalfen (Mads Ynddal) [Электронный ресурс] / Github. Режим доступа: <https://github.com/Baekalfen/PyBoy>, свободный. (дата обращения: 01.02.2022)
- [4] Cade Metz. TensorFlow, Google’s Open Source AI, Signals Big Changes in Hardware Too. [Электронный ресурс]: Wired, 2015 – Режим доступа: <https://www.wired.com/2015/11/googles-open-source-ai-tensorflow-signals-fast-changing-hardware-world>, свободный. (дата обращения: 02.02.2022)
- [5] Christian Kauten [Электронный ресурс] / Pypi. Режим доступа: <https://github.com/Baekalfen/PyBoy>, свободный. (дата обращения: 01.02.2022)
- [6] Erik D. Demaine. Tetris is Hard, Even to Approximate / Erik D. Demaine, Susan Hohenberger, David Liben-Nowell // [Электронный ресурс]: Arxiv, 2002 – Режим доступа: <https://arxiv.org/abs/cs/0210020>, свободный. (дата обращения: 11.12.2021)
- [7] Harvinder Singh. QRAS: efficient resource allocation for task scheduling in cloud computing / Harvinder Singh, Anshu Bhasin, Parag Ravikant Kaveri // [Электронный ресурс]: Researchgate, 2021 – Режим доступа: https://www.researchgate.net/publication/350192028_QRAS_efficient_resource_allocation_for_task_scheduling_in_cloud_computing, свободный. (дата обращения: 11.12.2021)
- [8] Kent, Steven. The Ultimate History of Video Games: From Pong to Pokemon: The Story Behind the Craze That Touched Our Lives and Changed the World (1st ed.). – New York, USA: Three Rivers Press, 2001. – С. 377-381.
- [9] Keras-RL [Электронный ресурс] / Github. Режим доступа: <https://github.com/keras-rl/keras-rl>, свободный. (дата обращения: 01.02.2022)
- [10] L. Langenhoven. Swarm tetris: Applying particle swarm optimization to tetris / L. Langenhoven, W. S. van Heerden, and A. P. Engelbrecht // [Электронный ресурс]: Ieeexplore, 2010 – Режим доступа: <https://ieeexplore.ieee.org/document/5586033>, свободный. (дата обращения: 11.12.2021)
- [11] Laura Wood. Global Cloud Computing Market (2020 to 2026) - by Service, Deployment, Application Type, End-user and Region [Электронный ресурс]: Businesswire, 2021 – Режим доступа: <https://www.businesswire.com/news/home/20210824005585/en/Global-Cloud-Computing-Market-2020-to-2026-by-Service-Deployment-Application-Type-End-user-and-Region-ResearchAndMarkets.com>, свободный. (дата обращения: 11.12.2021)
- [12] Matt Kapko. Can Huawei ‘Reinvent Itself’ as a Cloud Leader? [Электронный ресурс]: Sdxcentral, 2021 – Режим доступа: <https://www.sdxcentral.com/articles/news/can-huawei-reinvent-itself-as-a-cloud-leader/2021/04/>, свободный. (дата обращения: 11.12.2021)
- [13] nuno-faria, nlinker (Nick Linker) [Электронный ресурс] / Github. Режим доступа: <https://github.com/nuno-faria/tetris-ai>, свободный. (дата обращения: 11.12.2021)
- [14] OpenAI [Электронный ресурс] / Github. Режим доступа: <https://github.com/openai/gym>, свободный. (дата обращения: 01.02.2022)
- [15] Renan Samuel da Silva. Playing the Original Game Boy Tetris Using a Real Coded Genetic Algorithm / Renan Samuel da Silva, Rafael Stubs Parpinelli // [Электронный ресурс]: Researchgate, 2017 – Режим доступа: https://www.researchgate.net/publication/322321608_Playing_the_Original_Game_Boy_Tetris_Using_a_Real_Coded_Genetic_Algorithm, свободный. (дата обращения: 11.12.2021)
- [16] tensorflow [Электронный ресурс] / Github. Режим доступа: <https://github.com/tensorflow/tensorflow>, свободный. (дата обращения: 01.02.2022)

- [17] Tensorforce [Электронный ресурс] / Github. Режим доступа: <https://github.com/tensorforce/tensorforce>, свободный. (дата обращения: 01.02.2022)
- [18] Vihar Kurama. Обучение с подкреплением на языке Python / Vihar Kurama, Samhita Alla // [Электронный ресурс]: Habr, 2018 – Режим доступа: <https://habr.com/ru/company/piter/blog/434738>, свободный. (дата обращения: 02.02.2022)
- [19] X. Chen. Apply ant colony optimization to tetris / X. Chen, H. Wang, W. Wang, Y. Shi, and Y. Gao // [Электронный ресурс]: Dl, 2009 – Режим доступа: <https://dl.acm.org/doi/10.1145/1569901.1570136>, свободный. (дата обращения: 11.12.2021)

7 Приложения

Приложение 1

Ссылка на репозиторий проекта с исходным кодом и всеми использованными материалами.
<https://github.com/NikPeg/Reinforcement-learning-for-resource-allocation-tasks-in-the-cloud>

Приложение 2

Календарный план работ:

1. Завершить работу с тетрисом (31 января)
2. Изучение материалов по применению RL в облачных задачах, их описание в тексте курсовой работы, выбор конкретной практической задачи для решения, планирование эксперимента (28 февраля)
3. Реализация задачи в коде (31 марта)
4. Усложнение задачи: изменение условий среды или применение другой метрики (30 апреля)
5. Исправление ошибок в работе, оформление (20 мая)