

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
Факультет компьютерных наук Образовательная программа «Программная инженерия»  
(ВШЭ ФКН ПИ)**

УДК 004.852

**УТВЕРЖДАЮ**  
Академический руководитель  
образовательной программы  
«Программная инженерия»  
профессор департамента программной  
инженерии, канд. техн. наук  
\_\_\_\_\_ В.В. Шилов  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_г.

**ОТЧЕТ  
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

**ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ ДЛЯ ЗАДАЧ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В ОБЛАКЕ  
(заключительный)**

Руководитель НИР:  
Приглашенный лектор,  
главный инженер Huawei  
  
Андрей Александрович Тихонов  
\_\_\_\_\_ А. А. Тихонов  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_г.

Выполнил:  
Студент группы БПИ204  
образовательной программы  
«Программная инженерия»  
Пеганов Никита Сергеевич  
\_\_\_\_\_ Н. С. Пеганов  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_г.

Москва 2022

# 1 Реферат

Отчет 13 с., 1 кн., 4 рис., 2 табл., 19 источн., 1 прил. ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ, REINFORCEMENT LEARNING, RL, РАСПРЕДЕЛЕНИЕ РЕСУРСОВ В ОБЛАКЕ, ОБЛАЧНЫЕ ТЕХНОЛОГИИ, ОБЛАЧНЫЕ РЕСУРСЫ, TETRIS, OPENAI GYM, TENSORFLOW, KERASRL Объектом исследования являются особенности выделения ресурсов при работе с облачными сервисами. Целью работы является исследование применимости обучения с подкреплением в задачах распределения облачных ресурсов, а также сравнение данного подхода с другими методами решения задачи. В ходе работы проведен пробный эксперимент для изучения используемого метода — автоматическая игра в Тетрис. Также проведены экспериментальные исследования выбранного метода решения задачи. В процессе изучения иных работ было выяснено, что обучение с подкреплением применялось в решении задачи распределения данных в облаке, но данная область не является достаточно изученной. В результате проведенного исследования было выяснено, что выбранный мной способ не является применимым при решении данной задачи или требует дальнейших исследований. Исследование показало, что выбранный мной способ решения задачи не позволяет улучшить взятые для анализа метрики. Исходя из полученных результатов, можно предположить, что для улучшения полученных метрик возможно использовать другие виды машинного обучения.

## 2 Содержание

### Содержание

<b>1</b>	<b>Реферат</b>	<b>2</b>
<b>2</b>	<b>Содержание</b>	<b>3</b>
<b>3</b>	<b>Основные термины, определения и сокращения</b>	<b>4</b>
<b>4</b>	<b>Введение</b>	<b>5</b>
<b>5</b>	<b>Основная часть отчёта о НИР</b>	<b>6</b>
5.1	Применение обучения с подкреплением в игре «Тетрис» . . . . .	6
5.2	Применение обучения с подкреплением в основной задаче . . . . .	9
<b>6</b>	<b>Заключение</b>	<b>10</b>
<b>7</b>	<b>Приложения</b>	<b>13</b>

### 3 Основные термины, определения и сокращения

В настоящем отчете о НИР применяются следующие сокращения и обозначения. CPU (англ. central processing unit) — электронное устройство, исполняющее машинный код программ, главная часть аппаратного обеспечения компьютера. Иногда также называется микропроцессором или процессором ИТ (произносится ай-ти, сокращение от англ. Information Technology) — информационные технологии RAM (англ. Random Access Memory) — запоминающее устройство с произвольным доступом, один из видов памяти компьютера, позволяющий одновременно получить доступ к любой ячейке по её адресу на чтение или запись RL (англ. reinforcement learning) — обучение с подкреплением Облачные технологии — ИТ-технологии, которые позволяют хранить и обрабатывать информацию на удалённых от клиента серверах Облако — набор из некоторого количества ИТ-ресурсов, размещенных в инфраструктуре облачного провайдера Обучение с подкреплением (англ. reinforcement learning) — один из видов машинного обучения, в ходе применения которого испытываемая система, называемая агентом, обучается, взаимодействуя со средой

## 4 Введение

В первой части работы описано применение обучения с подкреплением для обучения агента самостоятельному прохождению в 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. Данная компьютерная игра выбрана неслучайно: она имеет концепции, сходные с основной задачей. Во-первых, ее основная цель — упаковка фигур. В решаемой задаче так же требуется распределять задачи пользователей между имеющимися ресурсами серверов. Во-вторых, игра имеет два параметра — координаты X и Y. Основная задача так же имеет два параметра, которые требуется распределять: CPU и RAM. Также решение задачи автоматической игры в "Тетрис" позволила научиться применять использованные библиотеки и фреймворки на практике.

### **Цели и задачи работы**

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

**Новизна и достоверность полученных результатов** Данная исследовательская работа не является первой работой, написанной на данную тему, однако, RL пока не было применено на практике для решения исследуемой задачи. Это показывает, что данная область нуждается в изучении.

### **Теоретическая значимость**

Работа не имеет теоретической значимости, так как опирается на уже исследованный алгоритм обучения с подкреплением.

### **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 Основная часть отчёта о НИР

### 5.1 Применение обучения с подкреплением в игре «Тетрис»

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"[6] 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[19] и Swarm tetris: Applying particle swarm optimization to tetris[10]. Примером использования RL для игры в Тетрис является статья A deep reinforcement learning bot that plays tetris[13].

Таким образом, исследование источников показало, что обучение с подкреплением активно используется для решения задачи автоматической игры в "Тетрис". Это объясняется тем, что "Тетрис"— дискретная задача, и в ней есть четкое разделение между средой и агентом. Средой выступает клетчатое поле, которое заполняется фигурами в процессе игры, а агентом является "игрокдвигающий фигуры влево и вправо. При этом агент взаимодействует со средой, расставляя на поле фигуры, а среда возвращает агенту количество удаленных строк и расположение поставленных фигур на поле.

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[3]. Однако он был отвергнут в пользу более популярной и более простой в использовании библиотеки gym-tetris[5], являющейся частью OpenAI Gym[14] — среды для симуляции известных компьютерных игр и физических задач.

При выборе библиотеки были рассмотрены pyqlearning[1] и Tensorforce[17]. Однако выбрана была библиотека KerasRL[9], надстройка над фреймворком TensorFlow[16]. Выбор был сделан в пользу KerasRL из-за совместимости со средой OpenAI Gym.

В игре "Тетрис" могут быть использованы различные метрики для расчета награды агента. Например, число убранных строк, количество сброшенных фигурок, число ходов до проигрыша, переход на новую скорость и другие. Для простоты в качестве награды было выбрано число убранных строк.

Рассмотрим библиотеки, выбранные для решения поставленной задачи. Библиотека OpenAI Gym применяется для обучения нейронных сетей игре в различные компьютерные игры, а также решения физических задач таких, как хождение и удержание баланса.

TensorFlow — фреймворк с открытым кодом, совмещающий в себе передовые достижения для создания и использования нейронных сетей. Библиотека создана компанией Google, однако активно развивается сообществом программистов[18].

Библиотека KerasRL позволяет создать нейронную сеть и агента, который будет взаимодействовать со средой и обучать сеть. Библиотека также является открытой и развивается компанией Google.

Для решения поставленной задачи выбран метод обучения с подкреплением. Это способ машинного обучения, при котором система, называемая агентом, обучается во время взаимодействия со средой (в первой части работы средой является компьютерная игра "Тетрис"). При этом агент влияет на среду с помощью действий, а среда взаимодействует с агентом, показывая ему информацию о состоянии среды, а также возвращая награду. Цель агента — максимизация награды.

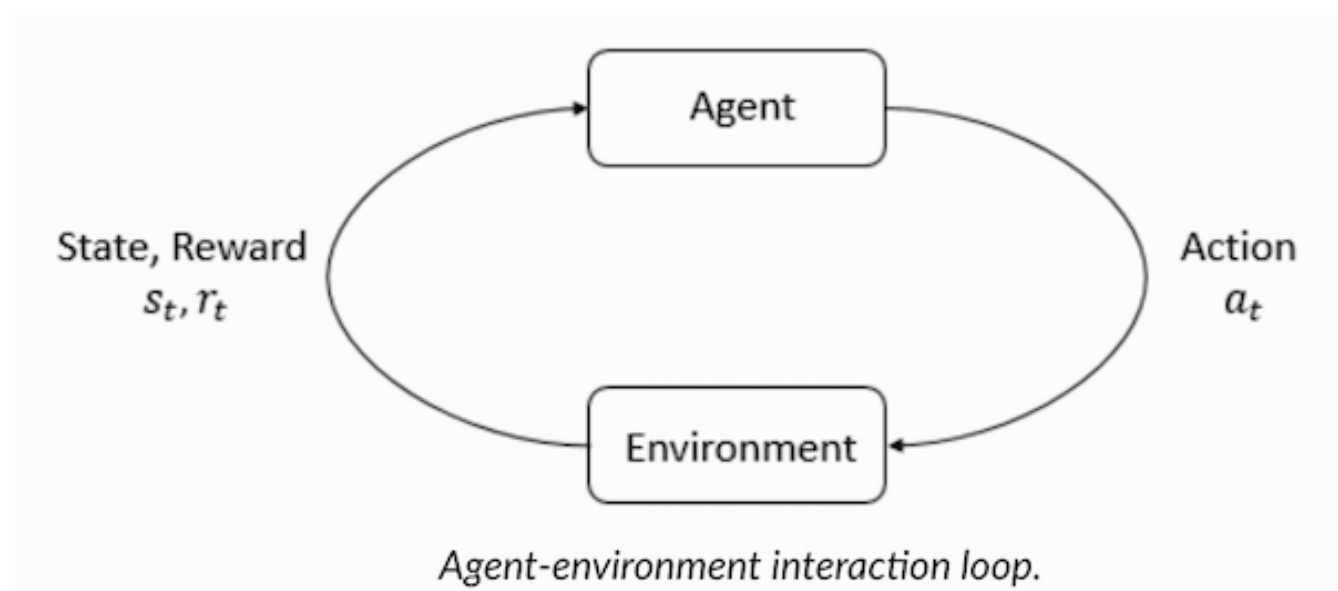


Рис. 1: Схема метода обучения с подкреплением[19].

В первую очередь, с помощью библиотеки OpenAI Gym была создана среда, в которой обучающийся агент производит какие-либо действия. В случае, рассматриваемом в данной работе, создается среда TetrisA-v0 — эмулятор игры "Тетрис". В этой среде агент может сделать ход, передав один из вариантов действий: сдвиг фигурки влево, сдвиг фигурки вправо, поворот на 90° по часовой стрелке, против часовой стрелки, ускорение падения фигурки. При заполнении строки частями фигурок среда автоматически очищает строку и добавляет 1 к награде.

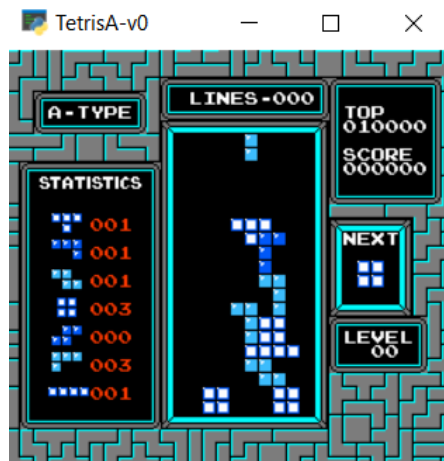


Рис. 2: Пример среды TetrisA-v0.

Затем с помощью библиотеки KerasRL была создана модель нейронной сети со следующими слоями. 1 слой, необходимый для сглаживания входных данных. Входными данными является информация, занята или свободна каждая клетка игрового поля. 2 сильно связанный слой с 24 входными синапсами и функцией активации выпрямленного линейного блока. Последний сильно связанный слой имеет линейную функцию активации и 12 выходных синапсов: каждый синапс отвечает за одно из действий.

Model: "sequential\_5"

Layer (type)	Output Shape	Param #
flatten_5 (Flatten)	(None, 184320)	0
dense_15 (Dense)	(None, 24)	4423704
dense_16 (Dense)	(None, 24)	600
dense_17 (Dense)	(None, 12)	300

=====  
Total params: 4,424,604  
Trainable params: 4,424,604  
Non-trainable params: 0  
=====

Рис. 3: Краткое описание нейронной сети.

На основе данной сети был создан агент с политикой BoltzmannQPolicy. Это означает, что агент вычисляет вид распределения значений, возвращаемых средой, и выбирает случайное действие на основе этого распределения.

После 50000 шагов, занявших 8 часов 11 минут, агент научился играть в "Тетрис" со средним значением награды 11.02. При этом среднее число шагов до проигрыша было равно 4260.23.

Сравним данный результат с другими реализациями RL в игре Tetris. В статье Playing the Original Game Boy Tetris Using a Real Coded Genetic Algorithm[7] средний результат алгоритма "FS-B" составил  $30.71 \pm 15.82$ , что является сравнимым с полученным значением. В работе Learn to Play Tetris with Deep Reinforcement Learning[20] алгоритмы Dreamer, Plan2xplore и LucidDreamer получили 55, 72 и 107 значений стертых линий, что является большим, но близким к полученному результату. Таким образом, эксперимент с автоматической игрой в "Тетрис" показал, что RL является применимым к данной задаче алгоритмом и получает значимый результат.



## 5.2 Применение обучения с подкреплением в основной задаче

## 6 Заключение

## Список использованных источников

- [1] *Kent, Steven* (2001) 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.). Three Rivers Press. С. 377-381.
- [2] *Arif Mohamed* (2018) A history of cloud computing // Сайт Computerweekly.com. 9 апреля (<https://www.computerweekly.com/feature/A-history-of-cloud-computing>) Просмотрено: 11.12.2021.
- [3] *Matt Kapko* (2021) Can Huawei ‘Reinvent Itself’ as a Cloud Leader? // Сайт Sdxcentral.com. 26 апреля (<https://www.sdxcentral.com/articles/news/can-huawei-reinvent-itself-as-a-cloud-leader/2021/04/>) Просмотрено: 11.12.2021
- [4] *Laura Wood* (2021) Global Cloud Computing Market (2020 to 2026) - by Service, Deployment, Application Type, End-user and Region Businesswire.com. 24 августа (<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>) Просмотрено: 11.12.2021
- [5] *Erik D. Demaine, Susan Hohenberger, David Liben-Nowell* (2002) Tetris is Hard, Even to Approximate // Сайт Arxiv.org. 21 октября (<https://arxiv.org/abs/cs/0210020>) Просмотрено: 11.12.2021
- [6] *Harvinder Singh, Anshu Bhasin, Parag Ravikant Kaveri* (2021) QRAS: efficient resource allocation for task scheduling in cloud computing // Сайт Researchgate.net. Апрель ([https://www.researchgate.net/publication/350192028\\_QRAS\\_efficient\\_resource\\_allocation\\_for\\_task\\_scheduling\\_in\\_cloud\\_computing](https://www.researchgate.net/publication/350192028_QRAS_efficient_resource_allocation_for_task_scheduling_in_cloud_computing)) Просмотрено: 11.12.2021
- [7] *Renan Samuel da Silva, Rafael Stubs Parpinelli* (2017) Playing the Original Game Boy Tetris Using a Real Coded Genetic Algorithm // Сайт Researchgate.net. Октябрь ([https://www.researchgate.net/publication/322321608\\_Playing\\_the\\_Original\\_Game\\_Boy\\_Tetris\\_Using\\_a\\_Real\\_Coded\\_Genetic\\_Algorithm](https://www.researchgate.net/publication/322321608_Playing_the_Original_Game_Boy_Tetris_Using_a_Real_Coded_Genetic_Algorithm)) Просмотрено: 11.12.2021
- [8] *X. Chen, H. Wang, W. Wang, Y. Shi, and Y. Gao* (2009) Apply ant colony optimization to tetris // Сайт Dl.acm.org. 8 июля (<https://dl.acm.org/doi/10.1145/1569901.1570136>) Просмотрено: 11.12.2021
- [9] *L. Langenhoven, W. S. van Heerden, and A. P. Engelbrecht* (2010) Swarm tetris: Applying particle swarm optimization to tetris // Сайт Ieeexplore.ieee.org. 18-23 июля (<https://ieeexplore.ieee.org/document/5586033>) Просмотрено: 11.12.2021
- [10] *nuno-faria, nlinker (Nick Linker)* (2019) A bot that plays tetris using deep reinforcement learning. // Сайт Github.com. 7 сентября (<https://github.com/nuno-faria/tetris-ai>) Просмотрено: 11.12.2021
- [11] *Baekalfen (Mads Ynddal)* (2021) Game Boy emulator written in Python. // Сайт Github.com. 22 октября (<https://github.com/Baekalfen/PyBoy>) Просмотрено: 01.02.2022
- [12] *Christian Kauten* (2019) An OpenAI Gym environment for Tetris on The Nintendo Entertainment System (NES) based on the nes-py emulator. // Сайт Pypi.org. 3 июня (<https://github.com/Baekalfen/PyBoy>) Просмотрено: 01.02.2022
- [13] *OpenAI* (2021) A toolkit for developing and comparing reinforcement learning algorithms. // Сайт Github.com. 2 октября (<https://github.com/openai/gym>) Просмотрено: 01.02.2022
- [14] *accel-brain, chimera0* (2020) Reinforcement Learning Library: pyqlearning. // Сайт Pypi.org. 13 июля (<https://pypi.org/project/pyqlearning/>) Просмотрено: 01.02.2022
- [15] *Tensorforce* (2021) Tensorforce: a TensorFlow library for applied reinforcement learning. // Сайт Github.com. 30 августа (<https://github.com/tensorforce/tensorforce>) Просмотрено: 01.02.2022
- [16] *Keras-RL* (2018) Deep Reinforcement Learning for Keras. // Сайт Github.com. 1 мая (<https://github.com/keras-rl/keras-rl>) Просмотрено: 01.02.2022
- [17] *tensorflow* (2021) An Open Source Machine Learning Framework for Everyone. // Сайт Github.com. 4 ноября (<https://github.com/tensorflow/tensorflow>) Просмотрено: 01.02.2022

- [18] *Cade Metz* (2015) TensorFlow, Google's Open Source AI, Signals Big Changes in Hardware Too. // Сайт Wired.com. 10 ноября (<https://www.wired.com/2015/11/googles-open-source-ai-tensorflow-signals-fast-changing-hardware-world/>) Просмотрено: 02.02.2022
- [19] *Vihar Kurama, Samhita Alla* (2018) Обучение с подкреплением на языке Python. // Сайт Habr.com. 28 декабря (<https://habr.com/ru/company/piter/blog/434738/>) Просмотрено: 02.02.2022
- [20] *Hanyuan Liu, Lixin Liu* (2020) Learn to Play Tetris with Deep Reinforcement Learning. // Сайт Openreview.net. 14 декабря (<https://openreview.net/forum?id=8TLyqLGQ7Tg>) Просмотрено: 05.05.2022

## 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>