

ISSN 2686-679X

ВЕСТНИК РГУ

Серия
«Информатика.
Информационная безопасность.
Математика»

Научный журнал

RSUH/RGGU BULLETIN

“Information Science.
Information Security. Mathematics”
Series

Academic Journal

Основан в 2018 г.
Founded in 2018

2
2022

VESTNIK RGGU. Seriya "Informatica. Informacionnaya bezopasnost. Matematica"

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" Series
Academic Journal

There are 4 issues of the printed version of the journal a year.

Founder and Publisher

Russian State University for the Humanities (RSUH)

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series is included: in the Russian Science Citation Index; in the List of leading scientific magazines journals and other editions for publishing PhD research findings peer-reviewed publications fall within the following research area:

20.00.00 Informatics

81.93.29 Information security, data protection

27.00.00 Mathematics

Objectives and areas of research

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series publishes the results of research by scientists from RSUH and other universities and other Russian and foreign academic institutions. The areas covered by contributions include theoretical and applied computer science, up-to-date IT, means and technologies of information protection and information security as well as the issues of theoretical and applied mathematics including analytical and imitation models of different processes and objects. Special emphasis is put on articles and reviews covering research in indicated directions in the areas of social and humanitarian problems and also issues of personnel training for these directions.

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series is registered by Federal Service for Supervision of Communications Information Technology and Mass Media. 25.05.2018, reg. No. FS77-72977

Editorial staff office: 6, Miusskaya sq., Moscow, Russia, 125047

tel: +7 (916) 250-90-85

e-mail: grnat@rambler.ru

ВЕСТНИК РГГУ. Серия «Информатика. Информационная безопасность. Математика»
Научный журнал

Выходит 4 номера печатной версии журнала в год.

Учредитель и издатель – Российский государственный гуманитарный университет (РГГУ)

ВЕСТНИК РГГУ, серия «Информатика. Информационная безопасность. Математика», включен: в систему Российского индекса научного цитирования (РИНЦ); в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук по следующим научным специальностям и соответствующим им отраслям науки:

20.00.00 Информатика

81.93.29 Информационная безопасность, защита информации

27.00.00 Математика

Цели и область

В журнале «Вестник РГГУ», серия «Информатика. Информационная безопасность. Математика», публикуются результаты научных исследований ученых и специалистов РГГУ, а также других университетов и научных учреждений России и зарубежных стран. Направления публикаций включают теоретическую и прикладную информатику, современные информационные технологии, методы, средства и технологии защиты информации и обеспечения информационной безопасности, а также проблемы теоретической и прикладной математики, включая разработку аналитических и имитационных моделей процессов и объектов различной природы. Особое внимание уделяется статьям и обзорам, посвященным исследованиям по указанным направлениям в области социальных и гуманитарных проблем, а также вопросам подготовки кадров по соответствующим специальностям для данных направлений.

ВЕСТНИК РГГУ, серия «Информатика. Информационная безопасность. Математика», зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций 25.05.2018 г., регистрационный номер ПИ № ФС77-72977.

Адрес редакции: 125047, Россия, Москва, Миусская пл., 6

Тел: +7 (916) 250-90-85

Электронный адрес: gnat@rambler.ru

Founder and Publisher

Russian State University for the Humanities (RSUH)

Editor-in-chief

V.V. *Arutyunov*, Dr. of Sci. (Computer Science), Russian State University for the Humanities (RSUH), Moscow, Russian Federation

Editorial Board

V.I. *Korolev*, Dr. of Sci. (Computer Science), professor, leading researcher, The Institute of Informatics Problems of the Russian Academy of Sciences (IPI RAN); professor Financial University under the Government of the Russian Federation, Moscow, Russian Federation (*deputy editor-in-chief*)

N.V. *Grishina*, Cand. of Sci. (Computer Science), associate professor, Russian State University for the Humanities (RSUH), Moscow, Russian Federation (*executive secretary*)

S.B. *Veprev*, Dr. of Sci. (Computer Science), professor, Russian Presidential Academy of National Economy and Public Administration, Moscow, Russian Federation

G.S. *Ivanova*, Dr. of Sci. (Computer Science), professor, Bauman Moscow State Technical University, Moscow, Russian Federation

V.M. *Maximov*, Dr. of Sci. (Physics and Mathematics), professor, Russian State University for the Humanities (RSUH), Moscow, Russian Federation

I.Yu. *Ozhigov*, Dr. of Sci. (Physics and Mathematics), professor, Lomonosov Moscow State University, Moscow, Russian Federation

E.A. *Primenko*, Cand. of Sci. (Physics and Mathematics), professor, Lomonosov Moscow State University, Moscow, Russian Federation

V.A. *Tsvetkova*, Dr. of Sci. (Computer Science), professor, Library for Natural Sciences of the RAS, Moscow, Russian Federation

Executive editor:

N.V. *Grishina*, Cand. of Sci. (Computer Science), associate professor, Russian State University for the Humanities (RSUH)

Учредитель и издатель

Российский государственный гуманитарный университет (РГГУ)

Главный редактор

В.В. Арутюнов, доктор технических наук, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация

Редакционная коллегия

В.И. Королев, доктор технических наук, профессор, ведущий научный сотрудник ФГУ «Федеральный исследовательский центр “Информатика и управление”» РАН, профессор, Финансовый университет при правительстве РФ, Москва, Российская Федерация (*заместитель главного редактора*)

Н.В. Гришина, кандидат технических наук, доцент, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация (*ответственный секретарь*)

С.Б. Венрев, доктор технических наук, профессор, Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации (РАНХиГС), Москва, Российская Федерация

Г.С. Иванова, доктор технических наук, профессор, Московский государственный университет им. Н.Э. Баумана, Москва, Российская Федерация

В.М. Максимов, доктор физико-математических наук, профессор, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация

И.Ю. Ожигов, доктор физико-математических наук, профессор, Московский государственный университет им. М.В. Ломоносова (МГУ), Москва, Российская Федерация

Э.А. Применко, кандидат физико-математических наук, профессор, Московский государственный университет им. М.В. Ломоносова (МГУ), Москва, Российская Федерация

В.А. Цветкова, доктор технических наук, профессор, Библиотека по естественным наукам РАН, Москва, Российская Федерация

Ответственный за выпуск:

Н.В. Гришина, кандидат технических наук, доцент, Российский государственный гуманитарный университет (РГГУ)

CONTENTS

Information Science

- Valerii V. Arutyunov*
Precarization. The effectiveness and relevance of results
of the scientific research in the second decade of the 21st century 8
- Mariya V. Vinogradova, Elizaveta A. Eliseeva,
Alekssei E. Samokhvalov*
Creating a cluster in a graph database Neo4j 18
- Kseniya V. Evdokimova, Kirill L. Tassov*
Analysis of algorithms built on “weak experts” 33

Mathematics

- Allaberdi G. Galkanov*
About some concepts and theorems in mathematical analysis 48
- Kirill E. Kazakov, Maksim A. Kudryavtsev*
MMFlow. Scalable flow-based model for image generation 59

СОДЕРЖАНИЕ

Информатика

- Валерий В. Арутюнов*
Прекаризация: результативность и востребованность
итогов научных исследований во втором десятилетии XXI века 8
- Мария В. Виноградова, Елизавета А. Алексеева,
Алексей Э. Самохвалов*
Создание кластера в графовой базе данных Neo4j 18
- Ксения В. Евдокимова, Кирилл Л. Тассов*
Анализ алгоритмов, построенных на «слабых экспертах» 33

Математика

- Аллаберди Г. Галканов*
О некоторых понятиях и теоремах математического анализа 48
- Кирилл Е. Казаков, Максим А. Кудрявцев*
MMFlow: масштабируемая потоковая модель
для генерации изображений 59

Информатика

УДК 331.5

DOI: 10.28995/2686-679X-2022-2-8-17

Прекаризация: результативность и востребованность итогов научных исследований во втором десятилетии XXI века

Валерий В. Арутюнов

*Российский государственный гуманитарный университет,
Москва, Россия, warut698@yandex.ru*

Аннотация. Рассматриваются особенности динамики изменения наукометрических показателей (публикационной активности, цитируемости и востребованности) результатов исследований российских специалистов в 2012–2021 гг. в области прекаризации; приводятся выявленные направления исследований в рассматриваемой области знаний, результаты работ по которым отличаются высокой востребованностью. В их числе: новые социальные группы в современных стратификационных системах глобального общества; занятость и социальная прекаризация в России; трансформация трудового правоотношения и новые формы занятости в условиях цифровой экономики; влияние неустойчивой занятости на переходы молодежи на рынке труда. Отмечается, что определяемый с учетом индекса Хирша уровень научной активности российских ученых в данной области исследований практически в 1,5 раза превышает минимальное значение мирового уровня научной активности ученого. Определены особенности изменения вышеуказанных показателей, включая практически непрерывный рост числа публикаций и показателя цитируемости практически до конца анализируемого периода; определены организации России, чьи итоги исследований в области прекаризации активно востребуются другими предприятиями страны. Анализ проводился на основе баз данных Российского индекса научного цитирования (РИНЦ).

Ключевые слова: прекаризация, публикационная активность, неустойчивая занятость, прекариат, наукометрические показатели, цитируемость, индекс Хирша

Для цитирования: Арутюнов В.В. Прекаризация: результативность и востребованность итогов научных исследований во втором десятилетии XXI века // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 2. С. 8–17. DOI: 10.28995/2686-679X-2022-2-8-17

© Арутюнов В.В., 2022

Precarization. The effectiveness and relevance of results of the scientific research in the second decade of the 21st century

Valerii V. Arutyunov

*Russian State University for the Humanities, Moscow, Russia,
warut698@yandex.ru*

Abstract. The features of the dynamics of changes in scientometric indicators (publication activity, citation and demand) of the research results of Russian specialists in 2012–2021 in the field of precarization are considered; the identified research directions in the field of knowledge under consideration are given, the results of which are in high demand. Among them: new social groups in modern stratification systems of global society; precariat: theory and empirical analysis (based on the materials of surveys in Russia, 1994–2013); employment and social precarization in Russia; transformation of labor relations and new forms of employment in the digital economy; the impact of unstable employment on youth transitions in the labor market. It is noted that the level of scientific activity of Russian scientists in the mentioned field of research, determined taking into account the Hirsch index, is 1.5 times higher than the minimum value of the world level of scientific activity of a scientist. The article reveals peculiarities of changes in the above indicators, including an almost continuous increase in the number of publications and the citation index almost until the end of the analyzed period and identifies the organizations of Russia whose results of research in the field of precarization are actively demanded by other enterprises of the country. The analysis was carried out on the basis of databases of the Russian Science Citation Index (RSCI).

Keywords: precarization, publication activity, precarious employment, precariat, scientometric indicators, citation, Hirsch index

For citation: Arutyunov, V.V. (2022), “Precarization. The effectiveness and relevance of results of the scientific research in the second decade of the 21st century”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 2, pp. 8–17, DOI: 10.28995/2686-679X-2022-2-8-17

Введение

К концу второго десятилетия XXI в. численность трудоспособного населения в Российской Федерации оценивалась на уровне около 80 млн человек (примерно 10 лет тому назад она составляла почти 88 млн); из них только 75 млн работают или ищут работу [Тощенко 2021]. По оценкам экспертов, этот тренд на снижение

численности трудоспособного населения России будет продолжаться до середины двадцатых годов XXI в.

Особую тревогу вызывает уменьшение численности работающей молодежи. Ее количество в возрасте 15–29 лет в последние годы уменьшалось следующим образом:

- в 2016 г. – на 789 тыс.;
- в 2017 г. – на 512,7 тыс.;
- в 2018 г. – на 839 тыс.;
- в 2019 г. – на 849,1 тыс. человек.

Пандемия, которая началась в 2019 г., нанесла серьезный удар по трудовой занятости населения страны. В 2021 г. численность трудоспособного населения сократилась до ~70,8 млн человек, а за 2020 г. – на 1,65 млн, из которых 1 млн – работники до 30 лет. К вышеизложенному необходимо добавить и рост безработицы, который вызывает тревогу – по официальным данным, ее уровень в 2021 г. составил ~6%.

В то же время российская экономика испытывает недостаток кадров, особенно в таких отраслях, как транспорт, лесное и рыбное хозяйство, строительство, занятость в которых составляет около 50%.

Среди основных проблем, связанных с развитием в России нестабильной трудовой занятости (прекаризации), выделяются следующие [Рахлис, Скворцова, Коптякова 2017]: дисбаланс спроса и предложения определенных категорий работников; значительное количество в стране моногородов, жизнь которых полностью зависит от работы одного-двух предприятий; недоверие граждан к социальным гарантиям из-за определенной нестабильности систем государственной политики; невысокий уровень заработных плат и, соответственно, низкий уровень платежеспособности населения.

Эти причины, во-первых, способствуют возникновению и развитию в стране социальной несправедливости и росту социальной напряженности в ряде субъектов Российской Федерации, где отмечается высокий уровень прекаризации; и, во-вторых, формированию в структуре общества нового феномена – класса прекариата (*людей, занятых на временных, сезонных и эпизодических работах, которые задействованы на неполный рабочий день или неполную ставку, выполняя краткосрочную или вспомогательную работу*), обострению противоречий между бедными и богатыми, усилению общественно-политической нестабильности. При этом в стране формируются две группы прекариата: добровольцы и вынужденный прекариат.

С учетом того факта, что за последние годы в этой предметной области стали активно публиковаться новые работы (например, [Бобков, Квачев, Новикова 2018; Кокшаров, Агарков, Сущенко

2020; Гоценко 2021; Черных 2019]), можно с определенной уверенностью сказать, что в настоящее время эта тема становится одной из достаточно актуальных не только в области труда и права, но и для других отраслей знаний, например, безопасности и социологии; это означает, что количество информации и результатов соответствующих исследований в этой предметной области знаний и смежных с ней отраслей будет только нарастать.

В связи с вышеизложенным возникает вопрос об интенсивности научных исследований российских специалистов в области прекаризации и их результатах, отраженных в публикациях, которые представляют интерес для научного сообщества и специалистов не только в области труда и права, но и для исследователей из других отраслей знаний.

Наукометрические показатели оценки результатов научных исследований

Как известно, с начала XXI в. в России все большее распространение получает оценка итогов исследовательской работы научных и образовательных организаций на базе наукометрических показателей результативности работ исследователей (публикационной активности P , индекса Хирша H и цитируемости C).

В настоящее время определенный интерес для исследователей во многих сферах наук представляют, кроме вышеуказанных наукометрических показателей, также востребованность V научным сообществом результатов научных исследований ученых, определяемая соотношением C/P .

Некоторые итоги анализа результатов исследований российских исследователей в области информационных технологий и по ряду других направлений развития науки и техники, полученных на основе баз данных РИНЦ (Российского индекса научного цитирования) [РИНЦ 2022], приводятся в ряде работ [Арутюнов, Гришина 2018; Арутюнов, Гришина 2020; Arutyunov 2016; Маршкова-Шайкевич 2008; Grinev 2019].

Ниже анализируется на основе данных РИНЦ динамика изменения в 2012–2021 гг. показателей публикационной активности, определяемой количеством публикаций P , их цитируемость C и востребованность V итогов работ российских исследователей в области прекаризации.

Динамика публикационной активности и цитируемости российских исследователей в сфере прекаризации в 2012–2021 гг. представлены на рис. 1.

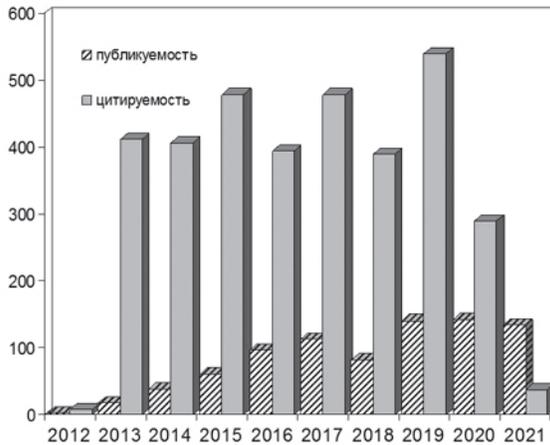


Рис. 1. Показатели публикационной активности и цитируемости российских исследователей в области прекаризации в 2012–2021 гг.

Как следует из рисунка, отмечается рост публикаций практически вплоть до конца анализируемого периода (несмотря даже на кризис 2014 г.). Этот факт свидетельствует о неослабевающем интересе российских ученых к анализируемой области исследований практически до конца второго десятилетия XXI в.; при этом в 2019–2021 гг. ежегодное количество публикаций стабилизировалось на уровне ~140 единиц, на что, возможно, в определенной степени повлияла и коронавирусная пандемия.

Что касается цитируемости, то ее максимум был достигнут в 2019 г., при этом в предыдущие шесть лет этот показатель стабилизировался на среднем значении этого показателя, равном 425.

Динамика востребованности итогов исследований российских ученых в области прекаризации за последние десять лет представлена на рис. 2.

Как видно из рис. 2, максимум показателя отмечался в 2013 г.; далее наблюдался спад его до минимума в 2021 г. в конце анализируемого периода, когда значение V уменьшилась почти в 10 раз по сравнению с максимумом. Незначительные показатели цитируемости и востребованности в 2021 г. объясняются, очевидно, известной причиной: замедленной по ряду причин «реакцией» научного сообщества в России на публикации текущего года.

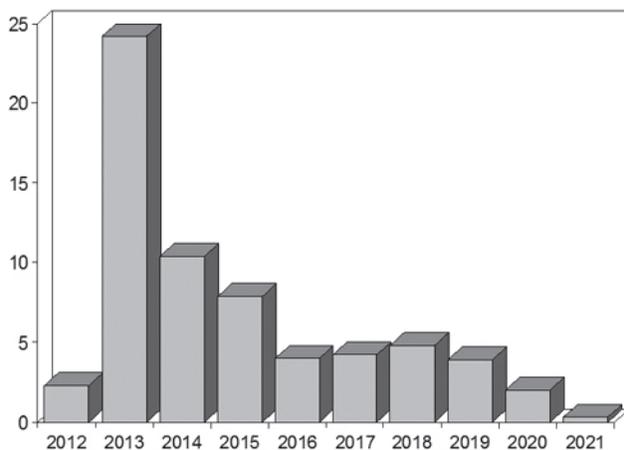


Рис. 2. Показатели востребованности в 2012–2021 гг. итогов работ российских исследователей в области прекаризации

Еще одним показателем, определяющим интенсивность развития научной деятельности в той или иной области наук, является индекс развития результативности исследовательской работы $I_d = I_{2p} / I_{1p}$, где I_{2p} и I_{1p} – суммарная публикационная активность исследователей за последние пять лет (I_{2p}) и пять предыдущих лет (I_{1p}) [Арутюнов 2017] – в нашем случае за 2017–2021 гг. и 2012–2016 гг. соответственно.

На рис. 3 представлены итоги сравнительного анализа индекса развития результативности исследовательской работы российских ученых для ряда направлений исследований: прекаризация, аутплейсмент (включающий комплекс мер, направленный на смягчение негативных последствий для персон и общества от увольнений сотрудников), информационная безопасность, квантовые технологии, профайлинг (включающий систему мер по прогнозированию и предупреждению криминальных действий) и медиации, под которой понимается достижение консенсуса конфликтующих сторон на основе их вступления в добровольные переговоры с использованием третьей стороны – посредника (медиатора).

Как видно из рис. 3, наибольшее значение индекса I_d отмечается для прекаризации, что свидетельствует о достаточно высокой значимости работ российских ученых в этой области знаний, привлекающей особенно в последние годы внимание научного сообщества.

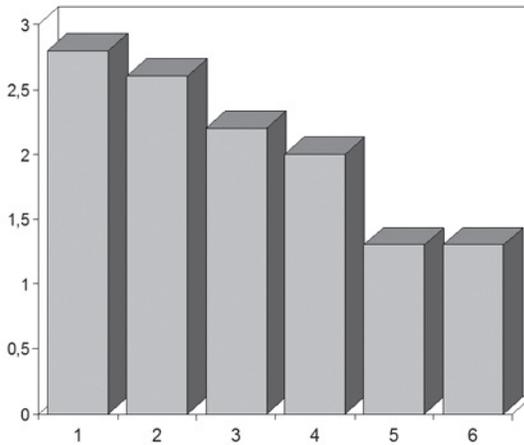


Рис. 3. Сравнительный анализ индексов развития результативности исследовательской работы для ряда направлений исследований:
 1 – прекаризация; 2 – профайлинг;
 3 – квантовые технологии; 4 – медиация;
 5 – информационная безопасность;
 6 – аутплейсмент

Следует отметить, что базы РИНЦ, аккумулируемые в Научной электронной библиотеке России уже более 15 лет, предоставляют исследователям возможность не только выявлять наукометрические показатели итогов работ. РИНЦ обладает развитым инструментарием, который позволяет на основе количественных наукометрических показателей определять не только уровень научной активности исследователей в определенной области знаний, но и наиболее востребованные итоги исследований, а также научную продуктивность исследователей-лидеров (организации и персоналии) в анализируемой области наук.

Так, выявленный для множества публикаций 2012–2021 гг. индекс Хирша в области прекаризации равен 25. Это достаточно высокое значение показателя свидетельствует, во-первых, о том, что при дальнейших исследованиях в анализируемой отрасли наук с большой вероятностью следует ожидать стабильную публикационную активность российских ученых, и, во-вторых, уровень их научной активности в данной области исследований практически в 1,5 раза превышает минимальный мировой уровень научной

активности ученого, равный 16 в соответствии с имеющимися рекомендациями [Ершова 2022]. Этот факт свидетельствует о высокой научной квалификации российских исследователей в области прекаризации.

В заключение необходимо заметить, что по итогам настоящей работы также удалось на основе анализа документального потока публикаций в области прекаризации с учетом наукометрических показателей определить направления работ в рассматриваемой сфере знаний, итоги работ по которым отличаются высокими показателями востребованности. В их число входят: теория и эмпирический анализ прекариата (на материалах опросов в России, 1994–2013 гг.); новые формы занятости в условиях цифровой экономики; занятость и социальная прекаризация в России; влияние прекаризации на переходы молодежи на рынке труда; новые социальные группы в современных стратификационных системах глобального общества.

В число выявленных организаций – флагманов в создании высоко востребованных результатов работ в области прекаризации – вошли Федеральный научно-исследовательский социологический центр РАН, МГУ им. М.В. Ломоносова, Национальный исследовательский университет «Высшая школа экономики», Московский государственный юридический университет им. О.Е. Кутафина.

Литература

- Арутюнов 2017 – *Арутюнов В.В.* Сравнительный анализ результативности научной деятельности в области информационных технологий и защиты информации // Материалы Международной конференции «Информация в современном мире». М.: ВИНТИ, 2017. С. 13–18.
- Арутюнов, Гришина 2018 – *Арутюнов В.В., Гришина Н.В.* Оценка результативности деятельности российских ученых: кластерный анализ (на примере естественно-научных отраслей) // Научные и технические библиотеки. 2018. № 9. С. 77–92.
- Арутюнов, Гришина 2021 – *Арутюнов В.В., Гришина Н.В.* Базовые региональные научные кластеры России // Вестник РГГУ, серия «Информатика. Информационная безопасность. Математика». 2021. № 2. С. 8–18.
- Бобков, Квачев, Новикова 2018 – *Бобков В.Н., Квачев В.Г., Новикова И.В.* Неустойчивая занятость в регионах Российской Федерации: результаты социологического исследования // Экономика региона. 2018. Т. 14. № 2. С. 366–379.
- Ершова 2022 – *Ершова С.К.* Инструкция по использованию РИНЦ. URL: <https://rf-gk.ru/profil-avtora-v-rinc-funktionalnye-vozmozhnosti-rossiiskii/> (дата обращения 24 февраля 2022).

- Кокшаров, Агарков, Сущенко 2020 – *Кокшаров В.А., Агарков Г.А., Сущенко А.Д.* Прекаризация труда как растущая форма занятости молодых специалистов в условиях пандемии // Экономика региона. 2020. Т. 16. № 4. С. 1061–1071.
- Маршакова-Шайкевич 2008 – *Маршакова-Шайкевич И.В.* Россия в мировой науке. М.: ИФРАН, 2008. 227 с.
- Рахлис, Скворцова, Коптякова 2017 – *Рахлис Т.П., Скворцова Н.В., Коптякова С.В.* Прекаризация как следствие трансформации социально-экономических систем // Вектор науки ТГУ. Серия: Экономика и управление. 2017. № 1 (28). С. 57–62.
- РИНЦ 2022 – *РИНЦ* (Российский индекс научного цитирования). URL: <https://elibrary.ru/querybox.asp?score=newquery> (дата обращения 24 февраля 2022).
- Тощенко 2021 – Прекарная занятость: истоки, критерии, особенности / Под ред. Ж.Т. Тощенко. М.: Весь мир, 2021. 400 с.
- Черных 2019 – *Черных Н.В.* Влияние нетипичных форм занятости на теоретические представления о трудовом отношении (на примере норм о дистанционном труде) // Актуальные проблемы российского права. 2019. № 8 (105). С. 108–117.
- Arutyunov 2016 – *Arutyunov V.V.* The results of priority research in the field of information security // Scientific and Technical Information Processing. 2016. Vol. 43. № 1. P. 42–46.
- Grinev – *Grinev A.V.* The use of scientometric indicators to evaluate publishing activity in modern Russia // Herald of the Russian Academy of Sciences. 2019. Vol. 89. № 5. P. 451–459.

References

- Arutyunov, V.V. (2016), “The results of priority research in the field of information security”, *Scientific and Technical Information Processing*, vol. 43, no. 1, pp. 42–46.
- Arutyunov, V.V. (2017), “Comparative analysis of the effectiveness of scientific activities in the field of information technology and information security”, *Materialy mezhdunarodnoi konferentsii «Informatsiya v sovremennom mire»* [Proceedings of the international conference “Information in the modern world”], VINITI, Moscow, Russia, pp. 13–18.
- Arutyunov, V.V. and Grishina, N.V. (2018), “Evaluation of performance of Russian scientists. Cluster analysis (by the example of the natural Sciences fields)”, *Nauchnye i tekhnicheskie biblioteki*, no. 9, pp. 77–92.
- Arutyunov, V.V. and Grishina, N.V. (2021), “Basic regional scientific clusters of Russia”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 2, pp. 8–18.
- Bobkov, V.N., Kvachev, V.G. and Novikova, I.V. (2018), “Unstable employment in the regions of the Russian Federation. Results of a sociological survey research”, *Economy of Regions*, vol. 14, no. 2, pp. 366–379.

- Chernykh, N.V. (2019), "Influence of atypical forms of the employment on theoretical ideas about labor relations (by the example of norms on distance labor)", *Actual problems of the Russian Law*, no. 8 (105), pp. 108–117.
- Ershova, S.K. (2022), "Instructions for using the RSCI", available at: <https://rf-gk.ru/profil-avtora-v-rinc-funkcionalnye-vozmozhnosti-rossiiskii/> (Accessed 24 February 2022).
- Grinev, A.V. (2019), "The use of scientometric indicators to evaluate the publishing activity in modern Russia", *Herald of the Russian Academy of Sciences*, vol. 89, no. 5, pp. 451–459.
- Koksharov, V.A., Agarkov, G.A. and Sushchenko, S.A. (2020), "Precarization of labor as a growing form of employment for young professionals in a pandemic", *Economy of Regions*, vol. 16, no. 4, pp. 1061–1071.
- Marshakova-Shaykevich, I.V. (2008), *Rossiia v mirovoi nauke* [Russia in the world science], RAS, Institute of Philosophy, Moscow, Russia.
- Rakhlis, T.P., Skvortsova, N.V. and Koptyakova, S.V. (2017), "Precarization as a consequence of the transformation in socio-economic systems", *Vector of Science TSU. Series: Economics and Management*, no. 1 (28), pp. 57–62.
- RSCI (2022), Russian Science Citation Index, available at: <https://elibrary.ru/querybox.asp?scope=newquery> (Accessed 24 February 2022).
- Toshchenko, J.T. (ed.) (2021), *Precarnaya zanyatost': istoki, kriterii, osobennosti* [Precarious employment. Origins, criteria, features], Ves' Mir, Moscow, Russia, 400 p.

Информация об авторе

Валерий В. Арутюнов, доктор технических наук, профессор, Российский государственный гуманитарный университет, Москва, Россия; 125047, Россия, Москва, Миусская пл., д. 6; warut698@yandex.ru

Information about the author

Valerii V. Arutyunov, Dr. of Sci. (Computer Science), professor, Russian State University for the Humanities, Moscow, Russia; bld. 6, Miusskaya Sq., Moscow, 125047, Russia; warut698@yandex.ru

Создание кластера в графовой базе данных Neo4j

Мария В. Виноградова

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
vinogradova.m@bmstu.ru*

Елизавета А. Елисеева

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
eliseeva.ea@outlook.com*

Алексей Э. Самохвалов

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
samox@bmstu.ru*

Аннотация. Цифровая трансформация всех отраслей народного хозяйства России требует от проектировщиков информационных систем с высокой скоростью обработки огромного числа запросов на предоставление и на обработку различных видов информации, «способности быстро адаптироваться к изменениям окружающей среды и оптимизировать свою работу “на лету”, подстраиваясь под ожидания клиента» [Мамаева 2018]. Для выполнения данного требования широко применяются кластеры баз данных. Каждый кластер представляет собой систему из одного ведущего сервера и нескольких ведомых серверов, между которыми настроена репликация данных. В статье предложена методика развертывания и настройки конфигурации кластера графовой базы данных Neo4j, приведен пример корректной репликации данных между экземплярами серверов и репликами, описан принцип разработки клиентского приложения для работы с кластером.

Материал данной работы может найти применение в информационных системах, построенных на основе облачных технологий хранения данных, что обеспечит высокую производительность, доступность, эластичную масштабируемость и быстрое развертывание [Анисифоров, Ильин 2012]. Развертывание кластера Neo4j возможно на современных платформах: AuraDB, Google Cloud Platform, Amazon EC2, Microsoft Azure и других.

Ключевые слова: кластеризация, репликация, кластер, графовая база данных, Neo4j

Для цитирования: Виноградова М.В., Елисеева Е.А., Самохвалов А.Э. Создание кластера в графовой базе данных Neo4j // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 2. С. 18–32. DOI: 10.28995/2686-679X-2022-2-18-32

Creating a cluster in a graph database Neo4j

Mariya V. Vinogradova

*Bauman Moscow State Technical University, Moscow, Russia,
vinogradova.m@bmstu.ru*

Elizaveta A. Eliseeva

*Bauman Moscow State Technical University, Moscow, Russia,
eliseeva.ea@outlook.com*

Alexei E. Samokhvalov

*Bauman Moscow State Technical University, Moscow, Russia,
samox@bmstu.ru*

Abstract. The digital transformation for all sectors of the national economy of Russia requires from designers of information systems a high speed of processing a huge number of requests for the provision and processing of various types of information, “the ability to quickly adapt to environmental changes and optimize their work “fleetingly”, adjusting to the client’s expectations” [Mamayeva 2018]. To meet that requirement, database clusters are widely used. Each cluster is a system of one master server and several slave servers, between which data replication is configured. The article proposes a technique for deploying and configuring a cluster of the Neo4j graph database, an example of correct data replication between server instances and replicas, and describes the principle of developing a client application for working with a cluster.

The material of the work can be used in information systems built on the basis of the cloud storage technologies, which will ensure high performance, availability, elastic scalability and rapid deployment [Anisiforov, Il’in 2012]. Deployment of a Neo4j cluster is possible on modern platforms: AuraDB, Google Cloud Platform, Amazon EC2, Microsoft Azure and others.

Keywords: clustering, replication, cluster, graph database, Neo4j

For citation: Vinogradova, M.V., Eliseeva, E.A. and Samokhvalov, A.E. (2022), “Creating a cluster in a graph database Neo4j”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 2, pp. 18–32, DOI: 10.28995/2686-679X-2022-2-18-32

Введение

В научной литературе можно встретить противоположные точки зрения на использование кластерной архитектуры информационной системы с применением графовой СУБД. В работе Н.Н. Шариповой, которая проводилась в рамках государственного задания Министерства образования и науки РФ в 2014–2016 гг., утверждается, что «так как в графовых моделях основной акцент сделан на связях, то такие БД более эффективно работают на одном сервере, чем на кластерах в распределенной сети» [Шарипова 2016]. Однако в исследованиях по проблемам хранения больших данных (расширение сети пользователей, растущая потребность в интерактивности, увеличение сложности систем) разработаны различные стратегии развертывания вычислительной инфраструктуры, для внедрения которых «использования одного сервера являются дорогостоящими и очень сложными, что вызвало сдвиг в сторону использования облачных аппаратных средств хранения для этой цели» [Чибисов, Шиманский, Клапатов, Чернявский 2020].

Авторы статьи провели исследование технологии работы с кластеризованными базами данных: создание хранилища, наполнения его множеством записей, организация кластера.

Цель работы – разработать методику развертывания кластера графовой базы данных и организации пользовательских запросов к нему.

Актуальность работы

От применения графовой СУБД ожидается не только значительный выигрыш в скорости доступа к данным и их обработки, но и в «удобстве использования, упрощении и ускорении процесса разработки и интеграции, снижении избыточности данных и количестве применяемых инструментов, а соответственно, и затрат на поддержку информационной системы в будущем» [Засядко, Карпов 2017]. Организация кластера на основе одного ведущего сервера и нескольких ведомых серверов представляет собой простое, но мощное решение проблемы.

Организация графовой базы данных

В качестве графовой системы управления базой данных выбрана Neo4j [Робинсон, Вебер, Эфрем 2016]. В данной СУБД выделяют

гибкую, простую и вместе с тем развитую модель данных, которую можно легко изменять, а результаты запросов (выборки) представляются в удобном для дальнейшего использования формате и очень просты в освоении. База не требует сложных объединений для извлечения связанных данных, так как достаточно получить сведения о соседнем узле или взаимосвязи без объединений или индексов.

Необходимо выбрать версию Neo4j для установки (дистрибутивы на сайте <https://neo4j.com>). Версия Community Edition – основа всего Neo4j – имеет открытый исходный код и распространяется под лицензией GPLv3; включает в себя фундаментальные функции базы данных, такие как типы данных, поддержку языка Cypher, смежность без индекса, полнотекстовый поиск, нативное хранение и обработка графов, нативное индексирование. Версия Enterprise Edition находится под коммерческой лицензией Neo4j. Добавляет расширенные операции и функции безопасности.

Версия Community Edition не обладает возможностью развертывания кластера, поэтому воспользуемся версией Enterprise Edition. Для начала настроим три экземпляра серверов (core server). Создаем три папки для хранения ядер (core server) и распаковываем в каждую по экземпляру СУБД Neo4j (в данном примере используется версия Neo4j Enterprise Edition 3.5.28 для Windows).

Перед развертыванием экземпляров серверов (core servers) необходимо их сконфигурировать. В каждой папке ядра в подкаталоге conf настраиваем файл neo4j.conf следующим образом:

- 1) `dbms.mode` – параметр, указывающий, в каком режиме будет работать сервер. Для настройки экземпляров серверов (core servers) этот параметр должен иметь значение “CORE”. Данный параметр будет совпадать для всех экземпляров (core-01, core-02, core-03);
- 2) `causal_clustering.minimum_core_cluster_size_at_formation` – параметр, указывающий минимальное количество экземпляров (core servers) для формирования кластера. Данный параметр будет совпадать для всех экземпляров (core-01, core-02, core-03);
- 3) `causal_clustering.minimum_core_cluster_size_at_runtime` – параметр, указывающий минимальное количество core servers, которое будет ожидаться для развертывания кластера. Данный параметр будет совпадать для всех экземпляров (core-01, core-02, core-03);
- 4) `causal_clustering.initial_discovery_members` – параметр, указывающий порты localhost, которые будут использовать экземпляры серверов (core servers). Данный параметр будет совпадать для всех экземпляров (core-01, core-02, core-03);

- 5) `causal_clustering.discovery_listen_address` – параметр, характеризующий экземпляры серверов (`core server`) для прослушивания его другими серверами (`core`) и репликами. В данном параметре можем указать только один порт. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 6) `causal_clustering.transaction_listen_address` – параметр, указывающий сетевой интерфейс и порт для прослушивания сервера доставки транзакций. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 7) `causal_clustering.raft_listen_address` – параметр, указывающий сетевой интерфейс для работы по протоколу Raft. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 8) `dbms.connector.bolt.listen_address` – параметр, указывающий сетевой интерфейс для работы по протоколу Bolt. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 9) `dbms.connector.http.listen_address` – параметр, указывающий сетевой интерфейс для работы по протоколу http. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 10) `dbms.connector.https.listen_address` – параметр, указывающий сетевой интерфейс для работы по протоколу https. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным;
- 11) `dbms.backup.listen_address` – параметр, указывающий сетевой интерфейс для работы с backup-сервером. Для каждого из экземпляров серверов (`core server`) параметр должен быть уникальным.

Заданные в работе параметры конфигурации приведены в табл. 1.

После того, как были сконфигурированы все экземпляры серверов (`core servers`), развернем каждый из них. Для этого откроем консоль Windows, перейдем в папку `bin` каждого экземпляра сервера (`core server`) и выполним команду `console`. Данную операцию нужно повторить с каждым из ядер серверов (`core server`). После можем проверить успешность развертывания серверов (`core server`) в консоли (рис. 1).

Таблица 1

Параметры для настройки конфигурации серверов

| | core-01 | core-02 | core-03 |
|---|----------------|----------------|----------------|
| <code>dbms.mode</code> | CORE | | |
| <code>causal_clustering_minimum_core_cluster_size_at_formation</code> | 3 | | |
| <code>causal_clustering_minimum_core_cluster_size_at_runtime</code> | 3 | | |
| <code>causal_clustering_initial_discovery_members</code> | localhost:5000 | localhost:5001 | localhost:5002 |
| <code>causal_clustering_discovery_listen_address</code> | :5000 | :5000 | :5002 |
| <code>causal_clustering_transaction_listen_address</code> | :6000 | :6001 | :6002 |
| <code>causal_clustering_raft_listen_address</code> | :7000 | :7001 | :7002 |
| <code>dbms.connector.bolt_listen_address</code> | :7687 | :7688 | :7689 |
| <code>dbms.connector.http_listen_address</code> | :7474 | :7475 | :7476 |
| <code>dbms.connector.https_listen_address</code> | :6474 | :6475 | :6476 |
| <code>dbms.backup_listen_address</code> | 0.0.0.0:6362 | 0.0.0.0:6363 | 0.0.0.0:6364 |

```

2021-06-02 11:05:48.049+0000 INFO ===== Neo4j 3.5.28 =====
2021-06-02 11:05:48.070+0000 INFO Starting...
2021-06-02 11:05:54.038+0000 INFO Initiating metrics...
2021-06-02 11:05:54.173+0000 INFO My connection info: [
  Discovery: listen=127.0.0.1:5001, advertised=localhost:5001,
  Transaction: listen=127.0.0.1:6001, advertised=localhost:6001,
  Raft: listen=127.0.0.1:7001, advertised=localhost:7001,
  Client Connector Addresses: bolt://localhost:7688,http://localhost:7475,https://localhost:6475
]
2021-06-02 11:05:54.176+0000 INFO Discovering other core members in initial members set: [localhost:5000, localhost
1, localhost:5002]
2021-06-02 11:05:56.817+0000 INFO Waiting for a total of 3 core members...
2021-06-02 11:06:06.189+0000 INFO Discovered core member at localhost:5000
2021-06-02 11:06:06.836+0000 INFO Waiting for a total of 3 core members...
2021-06-02 11:06:16.965+0000 INFO Waiting for a total of 3 core members...
2021-06-02 11:06:26.984+0000 INFO Waiting for a total of 3 core members...
2021-06-02 11:06:32.565+0000 INFO Discovered core member at localhost:5002
2021-06-02 11:06:36.295+0000 INFO Bound to cluster with id e9e7239a-7768-433d-8109-85463da3a652 as member id 641729
6ce-4c67-a29c-4c64a801888d
2021-06-02 11:06:55.319+0000 INFO Started downloading snapshot...
2021-06-02 11:07:29.783+0000 INFO Download of snapshot complete.
2021-06-02 11:07:57.867+0000 INFO Waiting to catchup with leader... we are 0 entries behind leader at 2.
2021-06-02 11:07:57.868+0000 INFO Successfully joined the Raft group.
2021-06-02 11:07:57.902+0000 INFO Sending metrics to CSV file at D:\neo4j-cluster\core-02\metrics
2021-06-02 11:07:59.678+0000 INFO Bolt enabled on 127.0.0.1:7688.
2021-06-02 11:08:00.951+0000 WARN Server thread metrics not available (missing neo4j.server.threads.jetty.all)
2021-06-02 11:08:00.958+0000 WARN Server thread metrics not available (missing neo4j.server.threads.jetty.idle)
2021-06-02 11:08:06.289+0000 INFO Started.
2021-06-02 11:08:06.988+0000 INFO Mounted REST API at: /db/manage
2021-06-02 11:08:07.203+0000 INFO Server thread metrics have been registered successfully
2021-06-02 11:08:10.732+0000 INFO Remote interface available at http://localhost:7475/

```

Рис. 1. Развертывание сервера core

В нашем примере ведущим сервером (leader core) стал core-02, а сервера core-01 и core-03 – ведомыми (follower). После деплоя кластера необходимо сконфигурировать множество реплик. Создаем три папки для каждого экземпляра neo4j, которые будут репликами, и повторяем аналогичную настройку (см. табл. 2).

Таблица 2

Параметры для настройки конфигурации реплик

| | replica-01 | replica-02 | replica-03 |
|---|----------------|----------------|----------------|
| <code>dbms.mode</code> | READ REPLICA | | |
| <code>causal_clustering.initial_discovery_members</code> | localhost:5000 | localhost:5001 | localhost:5002 |
| <code>causal_clustering.discovery_listen_address</code> | :5003 | :5004 | :5005 |
| <code>causal_clustering.transaction_listen_address</code> | :6003 | :6004 | :6005 |
| <code>dbms.connector.bolt_listen_address</code> | :7690 | :7691 | :7692 |
| <code>dbms.connector.http_listen_address</code> | :7477 | :7478 | :7479 |
| <code>dbms.connector.https_listen_address</code> | :6477 | :6478 | :6479 |
| <code>dbms.backup_listen_address</code> | 0.0.0.0:6365 | 0.0.0.0:6366 | 0.0.0.0:6367 |

Развертывание реплик (Read Replica) выполняется аналогично развертыванию экземпляров серверов (core servers).

Проверка корректности настройки построенного кластера выполняется командой `sysinfo` (язык Cypher, выполняется на одном из ядер, например core-02 (рис. 2) [Фаулер, Садаладж 2013]).

| Causal Cluster Members ? | | | | |
|--------------------------|---|--------|----------|---------|
| Roles | Addresses | Groups | Database | Actions |
| FOLLOWER | <code>bolt://localhost:7687, http://localhost:7474, https://localhost:6474</code> | - | default | Open |
| LEADER | <code>bolt://localhost:7688, http://localhost:7475, https://localhost:6475</code> | - | default | Open |
| FOLLOWER | <code>bolt://localhost:7689, http://localhost:7476, https://localhost:6476</code> | - | default | Open |
| READ_REPLICA | <code>bolt://localhost:7690, http://localhost:7477, https://localhost:6477</code> | - | default | Open |
| READ_REPLICA | <code>bolt://localhost:7691, http://localhost:7478, https://localhost:6478</code> | - | default | Open |
| READ_REPLICA | <code>bolt://localhost:7692, http://localhost:7479, https://localhost:6479</code> | - | default | Open |

Рис. 2. Состояние кластера

Добавим тестовые данные для того, чтобы проверить, корректно ли реплицируются данные между ядрами и репликами. Сначала перейдем к первому ядру по адресу <http://localhost:7474/browser/> (СУБД попросит авторизоваться. По умолчанию используем логин и пароль – neo4j/neo4j). Выполним команду для генерации данных Person на языке Cypher:

```
UNWIND range(0, 100) AS value
MERGE (person1:Person {id: value})
MERGE (person2:Person {id: toInteger(100.0 * rand())})
MERGE (person1)-[:FRIENDS]->(person2)
```

Проверим, что данные были реплицированы на ядрах и репликах. Второе ядро у нас расположено по адресу <http://localhost:7475/browser/>, первая реплика расположена по адресу <http://localhost:7477/browser/>. Выполним команду на выборку данных Person в количестве 10 экземпляров:

```
MATCH path = (person:Person)-[:FRIENDS]-(friend)
RETURN path
LIMIT 10
```

Используем существующий список “person” и связи. Возвращаем список и указываем ограничение результирующего множества. Как видно на рис. 3, данные присутствуют и в ядре, и в реплике, что позволяет сделать вывод о корректности репликации.

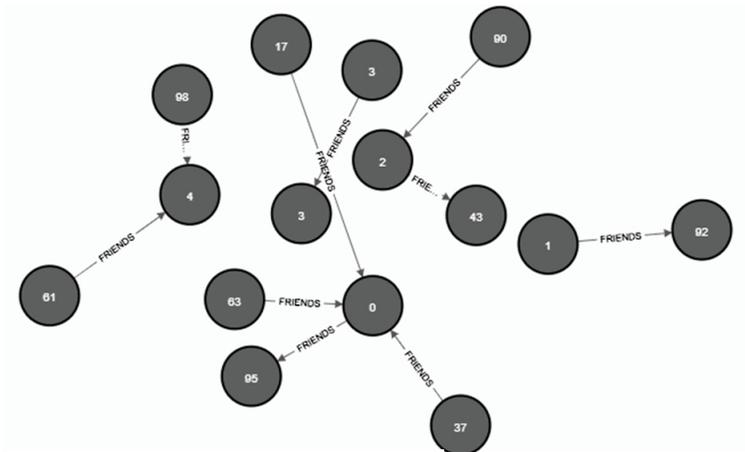


Рис. 3. Результат выполнения запроса выборки на ядре и реплике

Разработка приложения для работы с кластером

Авторы разработали приложение на языке Java, генерирующее тестовые данные. Использован популярный фреймворк Spring Boot и сборщик приложения Maven. Java-приложение состоит из модели, контроллера и представления. В данной работе модель представлена классами User и UserPermit, а контроллером выступает класс UserController.

В файл настройки сборщика pom.xml добавлена зависимость spring-boot-starter-data-neo4j для подключения драйвера для работы с базой данных:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-neo4j</artifactId>
  <version>${spring.version}</version>
</dependency>
```

Для того, чтобы подключиться к базе данных, необходим адрес ядра (URI), к которому выполняется соединение, и данные авторизации. В URI-адресе необходимо указывать только название ядра сервера, который является лидером (leader core), так как напрямую мы можем работать только с ним. Указываем адрес второго ядра (URI) и данные авторизации:

```
spring.neo4j.uri=bolt://localhost:7688
spring.data.neo4j.username=neo4j
spring.data.neo4j.password=neo4j
```

После установки зависимостей и конфигурирования к базе данных необходимо определить модели в базе данных на языке Java; создадим классы User и UserPermit.

Листинг кода модели класса User. Является отображением сущности в базе данных User:

```
@Data
@AllArgsConstructor
@Node
public class User {
  @Id
  @GeneratedValue
  private Long id;
```

```
private String name;
@Relationship(type = "Permit")
private List<UserPermit> userPermitList;
}
```

Листинг кода модели класса UserPermit, который является отображением сущности в базе данных UserPermit:

```
@Data
@QAllArgsConstructor
public class User Permit {
@Id
@GeneratedValue
private long id;
private String permit;
}
```

Для добавления данных в базу будем использовать паттерн Repository; для этого нужно определить интерфейс UserRepository:

```
public interface UserRepository extends Neo4jRepository<User,
Long> {}
```

Создадим класс UserService, который будет отделять логику репозитория от контроллера. В эксперименте необходим только метод insertUser для добавления пользователя в базу данных. Для транспортировки объекта с клиентской части используем паттерн DTO. Входящий объект UserDTO необходимо трансформировать в User. Эту функцию выполняет метод transform. Листинг кода, отражающий описание класса UserService:

```
@Service
@RequiredArgsConstructor(onConstructor_ = @Autowired)
public class UserService {
private final UserRepository userRepository;
public User insertUser(final userDT0) {return userRepository.
save(transform(userDT0));}
private User transform(final UserDT0 user) {
return new User(id: 0L,user.getName (),user.getUserPermitList());}}
```

Далее создаем контроллер, который будет принимать запрос на генерацию. В аннотации RequestMapping указываем контекст адреса для работы с этим контроллером. Метод insert генерирует

и добавляет в базу данных 1000 UserDTO и складывает результат добавления (идентификаторы записей id) в список.

Для того, чтобы обратиться к методу генерации, необходимо выполнить POST запрос на адрес <http://localhost:8080/users/generate>.

Листинг кода, отражающий описание класса UserController:

```
@Slf4j
@RestController
@RequestMapping("users")
@RequiredArgsConstructor(onConstructor_ = (@Autowired)
public class UsersController {
    private static final int COUNT = 1000;
    private final UserService userService;
    @PostMapping(«/generate»)
    public ResponseEntity<?> insert() {
        val listIdResult = new ArrayList<> ();
        for ( int i = 0; i < COUNT; i ++ ) {
            log.info(«GENERATE {} user», i);
            val userDTO = new UserDTO(
                UUID.randomUUID().toString(),
                UUID.randomUUID().toString(),
                userPermitList());
            userService.insertUser(userDTO);
            listIdResult.add(userDTO.getId());
        }
        return ResponseEntity.ok(listIdResult); }
}
```

Приложение готово. Теперь его необходимо запустить и выполнить POST-запрос, используя Postman (<https://www.postman.com>).

Для того, чтобы выполнить REST-запрос к java-серверу, необходимо указать адрес сервера, маппинг метода в контроллере и вид запроса (рис. 4).



Рис. 4. POST запрос на генерацию

Запрос на добавление 1000 записей был выполнен за 53932 мс. Теперь перейдем на одну из реплик, чтобы проверить, добавились

ли данные в базу. Переходим по адресу <http://localhost:7478/browser/> и выполняем команду:

```
MATCH path = (user:User)-[permit:Permit]-(friend)
RETURN path
LIMIT 8
```

Получаем список пользователей User и добавляем к ним связующие права, после чего возвращаем список и указываем ограничение результирующего множества. Результат выполнения запроса изображен на рис. 5. Для каждой вершины с меткой User было сгенерировано по две вершины с метками UserPermit, как и закладывалось в логику генерации вершин с метками User. Вывод: данные добавлены успешно.

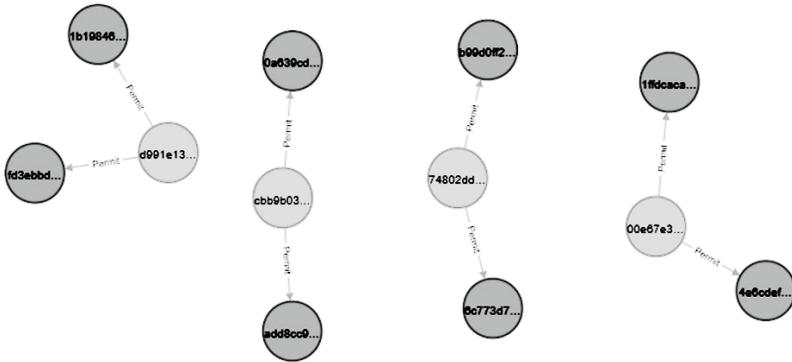


Рис. 5. Результат выполнения запроса

Заключение

Применение NoSql баз данных, и прежде всего графовых, позволяет хранить большие объемы полуструктурированной информации и предоставляет большие возможности по выполнению поисковых и аналитических запросов к ним. Это могут быть данные социальных сетей, электронные каталоги информационных ресурсов или семантические сети баз знаний [Виноградова, Ларионов, Черненький 2021]. Использование традиционных реляционных моделей для подобных структур данных было бы менее эффективно.

Организация баз данных в кластеры позволяет создавать большие хранилища информации, обеспечивает непрерывный до-

ступ к ним и сокращает время доступа к данным за счет обращения к репликам. Благодаря встроенным механизмам кластеризации сокращаются время и затраты на поддержку согласованности реплик и их синхронизацию, а также повышается уровень доступности данных.

Предложенная авторами методика развертывания кластера графовой базы данных будет полезна для построения различных информационных систем, в том числе гибридных интеллектуальных ИС [Черненко, Терехов, Гапанюк 2016], которые активно применяются для решения современных задач искусственного интеллекта.

Литература

- Анисифоров, Ильин 2012 – *Анисифоров А.Б., Ильин И.В.* Использование облачных технологий при построении информационных систем кластера // Экономика и управление. 2012. № 7. С. 22–27.
- Виноградова, Ларионов, Черненко 2021 – *Виноградова М.В., Ларионов А.С., Черненко В.М.* Интеллектуальная система для сбора и анализа информации о действиях пользователей асу предприятием // Динамика сложных систем – XXI век. 2021. Т. 15. № 2. С. 28–38.
- Засядко, Карпов 2017 – *Засядко Г.Е., Карпов А.В.* Проблемы разработки графовых баз данных // Инженерный вестник Дона. 2017. № 1 (44). С. 24.
- Мамаева 2018 – *Мамаева Г.А.* ИТ-решения для цифровой трансформации бизнеса // Эпоха науки. 2018. № 13. С. 81–86.
- Черненко, Терехов, Гапанюк 2016 – *Черненко В.М., Терехов В.И., Гапанюк Ю.Е.* Структура гибридной интеллектуальной информационной системы на основе метаграфов // Нейрокомпьютеры: разработка, применение. 2016. № 9. С. 3–13.
- Фаулер, Садаладж 2013 – *Фаулер М., Садаладж П.Д.* NoSQL: новая методология разработки нереляционных баз данных: Пер. с англ. М.: ООО «И.Д. Вильямс», 2013. С. 192 с.
- Чибисов, Шиманский, Клапатов, Чернявский 2020 – *Чибисов И.В., Шиманский В.В., Клапатов И.А., Чернявский Ю.А.* NoSQL – решение проблем хранения больших данных // Big Data and Advanced Analytics. 2020. № 6. С. 340–345.
- Робинсон, Вебер, Эфрем 2016 – *Робинсон Я., Вебер Д., Эфрем Э.* Графовые базы данных: новые возможности для работы со связанными данными / Пер. с англ. Р.Н. Рагимова; науч. ред. А.Н. Кисилев. М.: ДМК Пресс, 2016. 256 с.
- Шаринова 2016 – *Шаринова Н.Н.* Об использовании NOSQL-хранилищ данных // Восточно-Европейский научный журнал. 2016. Т. 9. № 3. С. 73–76.

References

- Anisiforov, A.B. and Il'in I.V. (2012), "The use of cloud technologies in the construction of cluster information stems", *Ekonomika i upravleniye*, no. 7, pp. 22–27.
- Chernen'kii, V.M., Terekhov, V.I. and Gapanyuk, Yu.E. (2016), "The structure of the hybrid intelligent information system based on metagraphs", *Neirokomputers*, no. 9, pp. 3–13.
- Chibisov, I.V., Shimanskii, V.V., Klapatok, I.A. and Chernyavskii, Yu.A. (2020), "NoSQL – solving big data storage issues", *Big Data and Advanced Analytics*, no. 6, pp. 340–345.
- Fauler, M. and Sadaladzh, P.D. (2013), *NoSQL: novaya metodologiya razrabotki nerelyatsionnykh baz dannykh* [NoSQL: A New Methodology for Non-Relational Database Development], ООО "I.D.Vil'yame", Moscow, Russia.
- Mamayeva, G.A. (2018), "IT solutions for digital transformation of business", *Epokha nauki*, no. 13, pp. 81–86.
- Robinson, Ya., Veber, D. and Eifrem, E. (2016), "Графовые базы данных: новые возможности для работы со связанными данными", [Graph databases: new possibilities for working with related data], Ragimov, R.N. (transl. from English); Kiselev, A.N. (ed.), DMK Press, Moscow, Russia.
- Sharipova, N.N. (2016), "On the use of NOSQL data warehouses, *Vostochno-Yevropeyskiy nauchnyy zhurnal*, T. 9, no. 3, pp. 73–76.
- Vinogradova, M.V., Larionov, A.S. and Chernen'kii, V.M. (2021), "An intelligent system for an enterprise collecting and analyzing information about the actions of automated control system users", *Dynamics of Complex Systems – XXI Century*, vol. 15, no. 2, pp. 28–38.
- Zasyadko, G.Ye. and Karpov, A.V. (2017), "Issues of the development of graph databases", *Inzhenernyy vestnik Dona*, vol. 1 (44), p. 24.

Информация об авторах

Мария В. Виноградова, кандидат технических наук, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; vinogradova.m@bmstu.ru

Елизавета А. Елисеева, магистрант, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; eliseeva.ea@outlook.com

Алексей Э. Самохвалов, кандидат экономических наук, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; samox@bmstu.ru

Information about the authors

Mariya V. Vinogradova, Cand. of Sci. (Computer Sciences), Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Bauman Str., Moscow, Russia, 105005; vinogradova.m@bmstu.ru

Elizaveta A. Eliseeva, master student, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Bauman Str., Moscow, Russia, 105005; eliseeva.ea@outlook.com

Alexei E. Samokhvalov, Cand. of Sci. (Economics), Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Bauman Str., Moscow, Russia, 105005; samox@bmstu.ru

Анализ алгоритмов, построенных на «слабых экспертах»

Ксения В. Евдокимова

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
ks.evdokimova@mail.ru*

Кирилл Л. Тассов

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
ktassov@policiesoft.ru*

Аннотация. Машинное обучение – достаточно сложный процесс, поскольку для его реализации необходимо оптимизировать обучающий набор данных, время и качество обучения. Достичь данную оптимизацию в ряде случаев сложно и зачастую для этого используется ансамблевое обучение. При ансамблевом обучении можно получать прогнозы из нескольких источников или классификаторов, основываясь на надежности и опыте каждого источника. В результате задача такого обучения состоит в постройке надежной системы классификаторов с целью получения точного прогноза. В статье дается краткое введение в ансамбли экспертов и их применение в машинном обучении; рассматривается разработка, внедрение и применение таких алгоритмов, в частности, исследуются условия, при которых системы на основе ансамблей могут быть более действенными, чем их аналоги на основе одного классификатора. Приводится классификация алгоритмов, основанных на «слабых экспертах», а также описываются методики работы, полученные из них. Описываются преимущества и недостатки каждого из видов ансамблей «слабых экспертов». Построена сравнительная таблица данных ансамблей с учетом необходимого объема обучающей выборки, вида используемых в ансамбле экспертов и процента распознавания.

Ключевые слова: «слабые эксперты», бэггинг, бустинг, стэкинг, ансамбли, AdaBoost, bagging decision trees, Straight Stack

Для цитирования: Евдокимова К.В., Тассов К.Л. Анализ алгоритмов, построенных на «слабых экспертах» // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 2. С. 33–47. DOI: 10.28995/2686-679X-2022-2-33-47

© Евдокимова К.В., Тассов К.Л., 2022

Analysis of algorithms built on “weak experts”

Kseniya V. Evdokimova

*Bauman Moscow State Technical University, Moscow, Russia,
ks.evdokimova@mail.ru*

Kirill L. Tassov

*Bauman Moscow State Technical University, Moscow, Russia,
ktassov@policesoft.ru*

Abstract. Machine learning is a rather complex process, since its implementation requires optimization of the training dataset, time and quality of training. In some cases, that optimization is difficult to achieve, and ensemble learning is often used. Ensemble learning can generate predictions from multiple sources or classifiers based on the reliability and experience of each source. As a result, the task of such learning is to build a system of classifiers in order to obtain an accurate forecast. The article introduces the ensembles of experts in machine learning and considers the development, introduction and application of such algorithms, in particular, examines the conditions under which systems based on ensembles can be more effective than their counterparts based on a single classifier. A classification of algorithms based on “weak experts” is given, and the methods used by the main ones are described. The article also specifies the advantages and disadvantages of each type of “weak expert” ensembles. A comparative table of ensembles data was built, taking into account the required amount of training sample, the type of experts used in the ensemble and the percentage of recognition.

Keywords: “weak experts”, bagging, boosting, stacking, ensembles, Ada-Boost, bagging decision trees, Straight Stack

For citation: Evdokimova, K.V. and Tassov, K.L. (2022), “Analysis of algorithms built on ‘weak experts’”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 2, pp. 33–47, DOI: 10.28995/2686-679X-2022-2-33-47

Введение

Основная задача науки – получение правильных предсказаний о будущем поведении сложных систем на основании их прошлого поведения. Многие задачи, возникающие в практических приложениях, не могут быть решены заранее известными методами и алгоритмами с минимальным процентом ошибки. Происходит это по нескольким причинам: имеется недостаточный объем обучающей выборки, попадание в локальные минимумы при обучении, область

решения задач превосходит область определения метода. Таким образом, мы приходим к необходимости объединения нескольких экспертов. Алгоритм принятия решения, построенный на нескольких экспертах, носит различные названия: система множественного классификатора, ансамбль экспертов, смесь экспертов [Hawelka, Sitko, Kazakoroulos, Beinat 2017].

Идея ансамблевого обучения состоит в том, чтобы организовать такой пул экспертов, который позволит объединить их в общую систему, игнорируя монологичную. В качестве эксперта могут рассматриваться различные методы предсказания, стохастические теории, методы регрессии и т. д. Суть ансамблевых методов состоит в том, что эксперты предоставляют свои прогнозы прежде, чем представляется соответствующий исход. При получении каждого очередного исхода каждый из экспертов вычисляет свои потери, которые в последующем может использовать агрегирующий алгоритм [Yu, Luo, Wu, Han, You, Leung, Zhang 2016, Zhou 2012].

В этом контексте возникают два вопроса: первый касается проблемы создания пула разнообразных экспертов, а второй заключается в том, каким именно образом объединить экспертов для достижения максимальной точности общей системы. Изучение ансамблей стало популярным методом машинного обучения в течение последних 30 лет. За это время было предложено большое количество алгоритмов для решения различных задач, потому и появилась необходимость собрать и классифицировать их.

Итак, все ансамблевые алгоритмы имеют общую архитектуру. Ансамбль содержит несколько экспертов, называемых базовыми. Базовые эксперты генерируются на основе обучающих данных, и их результаты объединяются с помощью некоторого базового алгоритма, который может быть представлен в разных формах: дерево решений, нейронная сеть или другие.

Обобщающая способность ансамбля часто намного сильнее, чем у базовых экспертов. Ансамблевые методы привлекательны в основном потому, что они способствуют усилению «слабых экспертов», которые, в свою очередь, могут ненамного превосходить случайные догадки «сильных экспертов», дающих более точные прогнозы. Таким образом, базовые эксперты также могут называться «слабыми экспертами».

Наиболее часто ансамблевые алгоритмы группируются следующим образом:

1. По цели объединения экспертов в систему:

- 1.1 «бэггинг» (англ. bagging) – такие алгоритмы уменьшают дисперсию: выделяется несколько обучающих наборов из единого и на них обучаются независимые эксперты;

- 1.2 «бустинг» (англ. boosting) – алгоритмы бустинга заключаются в минимизации смещения, таким образом, каждый последующий эксперт покрывает недостатки предыдущего;
 - 1.3 «стэкинг» (англ. stacking) – в данном случае целью алгоритмов является улучшение прогнозирования за счет распределения метапризнаков по уровням.
 2. По расположению экспертов в ансамблях:
 - 2.1 последовательные ансамблевые алгоритмы – базовые эксперты генерируются последовательно, основная черта – использование зависимости между базовыми экспертами;
 - 2.2 параллельные ансамблевые алгоритмы – базовые эксперты генерируются параллельно, основная черта – базовые эксперты независимы.
 3. По природе экспертов в ансамблях:
 - 3.1 гомогенные ансамбли – эксперты используют единый базовый алгоритм обучения для создания однородных базовых экспертов, то есть экспертов одного типа;
 - 3.2 гетерогенные ансамбли – используются эксперты различных типов, что позволяет повысить точность всего ансамбля.
- Обратимся к рассмотрению основных ансамблевых алгоритмов с учетом степени их использования в решении тех или иных задач.

«Бустинг»-алгоритмы

Большим количеством алгоритмов отличается группа «бустинг» алгоритмов. Термин «бустинг» (с англ. – «увеличение») отражает основную сущность этих алгоритмов: это семейство алгоритмов, которые преобразуют несколько «слабых экспертов» в единого «сильного эксперта». Интуитивно «слабый эксперт» лишь немного лучше, чем случайная догадка «сильного эксперта», но их объединение позволяет получить «сильного эксперта», близкого к совершенству.

Вопрос, который породил появление этих алгоритмов, был поставлен Кернсом и Валиантом в 1989 г. [Zhou 2012]. Вопрос был поставлен так: равны ли два класса сложности для слабо и сильно обучаемых задач? Этот вопрос имеет принципиальное значение, поскольку если ответ положительный, любой «слабый эксперт» потенциально может быть усилен до «сильного эксперта», особенно учитывая, что на практике, как правило, очень легко получить «слабых экспертов», но трудно получить сильных. В 1990 г. Шапир (Robert Schapire) доказал истинность поставленного вопроса.

В одной из своих первых работ [Schapire 1990] Шапир приводит описание алгоритма AdaBoost, появление которого и связано

с доказательством истинности вопроса. AdaBoost представляет собой процесс, включающий несколько подпроцессов, каждый из которых должен иметь эксперта. Этот алгоритм пытается построить лучшую модель, используя экспертов, представленных в подпроцессах.

AdaBoost (Adaptive Boosting) – мета-алгоритм и может совместно использоваться со многими другими алгоритмами обучения для увеличения их производительности. AdaBoost адаптивен в том смысле, что последующие классификаторы настраиваются в пользу тех экземпляров, которые классифицируются по предыдущим классификаторам. AdaBoost чувствителен к шумным данным и выбросам. Однако в некоторых проблемах он может быть менее восприимчивым к переобучению, чем другие алгоритмы.

AdaBoost определяет для каждой из серий своего эксперта и во время каждого вызова обновляется распределение весов, которое указывает на важность результатов каждого из экспертов для общего результата. При каждой итерации веса каждого неправильно классифицированного примера увеличиваются, а правильно классифицированного – уменьшаются, именно этим и достигается фокусировка каждого следующего эксперта на примерах, ускользающих от правильной классификации.

AdaBoost относится к классу алгоритмов AnyBoost. Данный класс был разработан посредством обобщения таких методов, как AdaBoost, Мейсоном и Френом в 1999 г. Все алгоритмы, относящиеся к этому классу, объединяет вид: алгоритмы ускорения градиентного спуска, которые объединяют «слабых экспертов» для минимизации функции потерь. Настройку модели можно представить следующим образом: $X * Y$ – пространство выборки, где X – пространство признаков и Y – пространство меток. $P = P(X * Y)$ – множество всех вероятностных мер на $X * Y$. F – семейство классификаторов f («слабые эксперты»), для которых $f: X \in Y$. Обозначим через $\text{lin}(F)$ – множество всех линейных комбинаций элементов в F с внутренним произведением h, i . Функцию потерь обозначим через L : $\text{lin}(F) \in R$. Цель состоит в нахождении

$$F \in \text{lin}(F) : L(F) = \min, \quad (1)$$

т. е. необходимо решить

$$F^* = \underset{F \in \text{lin}(F)}{\text{argmin}} L(F). \quad (2)$$

Основная идея алгоритмов AnyBoost состоит в том, что для $F \in \text{lin}(F)$ требуется найти такое $f \in F$, что $L(F + \xi f) - L(F)$ от-

рицательно, т. е. $L(F + \xi F)$ уменьшается. Поэтому естественным выбором f будет $f = -\nabla(L(F))$, но в целом весьма вероятно, что $-\nabla L(F) \notin F$. По разложению Тейлора $L(F + \xi F) = L(F) + \xi \langle \nabla L(F), f \rangle + o(\xi)$ и, следовательно, для минимизации $L(F + \xi F) - L(F)$ хорошим выбором f будет тот, который максимизирует $-\langle \nabla L(F), f \rangle$ [Wan 2017].

Еще одним алгоритмом класса AnyBoost является LogitBoost. LogitBoost работает аналогично AdaBoost, за исключением того, что сводит к минимуму биномиальное отклонение

$$\sum_{n=1}^N w_n \log(1 + \exp(-2y_n f(x_n))), \quad (3)$$

где $y_n \in \{-1, +1\}$ – это настоящая метка класса, w_n – веса наблюдений, нормированные для суммирования до 1, $f(x_n) \in \{-\infty, +\infty\}$ – прогнозируемая классификационная оценка.

Биномиальное отклонение придает меньший вес плохо классифицированным объектам (объекты с большими отрицательными значениями $y_n f(x_n)$). LogitBoost может дать лучшую среднюю точность, чем AdaBoost для данных с плохо разделяемыми классами.

Подгонка регрессионной модели на каждом этапе повышения превращается в большое вычислительное преимущество для данных с многоуровневыми категориальными предикторами. Возьмите категорический предиктор с L -уровнями. Чтобы найти оптимальное разделение решений по такому предиктору, в дереве классификации необходимо учитывать разбиения $2^{L-1} - 1$. Дерево регрессии должно учитывать только $L-1$ разбиения, поэтому время обработки может быть намного короче. LogitBoost рекомендуется для категориальных предикторов со многими уровнями.

Еще одним представителем «бустинг» алгоритмов является LPBoost (ускорение линейного программирования). Данный алгоритм выполняет мультиклассовую классификацию, пытаясь максимизировать минимальный запас в обучающем наборе. Эта попытка использует алгоритмы оптимизации, а именно линейное программирование.

Запас классификации – это разница между прогнозируемой оценкой мягкой классификации для истинного класса и наибольшей оценкой для ложных классов. Для деревьев оценка классификации листового узла является апостериорной вероятностью классификации в этом узле. Апостериорная вероятность классификации в узле – это количество обучающих последовательностей, которые ведут к этому узлу с классификацией, деленное на количество обучающих последовательностей, которые ведут к этому узлу.

LPBoost итеративно максимизирует минимальный запас через последовательность задач линейного программирования. Эквивалентно по двойственности LPBoost минимизирует максимальное ребро, где ребро – это взвешенное среднее поле. На каждой итерации в задаче есть больше ограничений. Таким образом, для больших задач проблема оптимизации становится все более сложной и медленной для решения.

LPBoost обычно создает ансамбли со многими учениками, вес которых на порядки меньше, чем у других учеников. Поэтому, чтобы лучше разрешить удаление неважных членов ансамбля, компактный метод изменяет порядок членов ансамбля LPBoost с наименьшего веса на наименьший. Поэтому вы можете легко удалить наименее важные члены ансамбля.

«Бэггинг» алгоритмы

Следующим рассматриваемым типом алгоритмов будет Bagging. Название bagging произошло из аббревиатуры Bootstrap AGGregatING. Автором данной идеи стал Брейман (Breiman) в 1996 г. [Breiman 1996]. Сутью данных алгоритмов является знание о том, что сочетание базовых экспертов приводит к резкому уменьшению ошибки, а потому необходимо, чтобы эксперты были максимально независимыми. Решений данной задачи несколько. На поверхности лежит идея учитывать обучающие данные, т. е. выбирать их таким образом для различных экспертов, чтобы они не перекрывали друг друга. Однако мы ограничены обучающими данными (не существует бесконечного набора) и, следовательно, такой процесс породит маленькие и непрезентативные образцы, в результате чего базовые эксперты будут малоэффективны.

Обычная стратегия для комбинации деревьев решений основана на создании нескольких деревьев решений, объединенных с критерием большинства голосов. Если появляется неклассифицированный экземпляр, то каждый отдельный классификатор делает прогноз, и этому экземпляру присваивается значение класса с наибольшим количеством голосов.

«Бэггинг» Бреймана [Khoshgoftaar, Van Hulse, Napolitano 2011] (или Bootstrap Aggregating) – это интуитивно понятный и простой метод, который демонстрирует хорошую производительность, уменьшает дисперсию и избегает перегрузки. Обычно этот алгоритм реализуется с помощью деревьев решений, но может применяться с любым типом классификатора.

Разнообразие в наборе достигается путем создания реплицированных наборов данных начальной загрузки исходного набора обучающих данных:

различные наборы обучающих данных выбираются случайным образом с заменой из исходного обучающего набора, и, следовательно, реплицированные наборы обучающих данных имеют тот же размер, что и исходные данные, но некоторые экземпляры могут не появляться в нем или могут появляться более одного раза [Breiman 1996].

После этого для каждого нового экземпляра набора обучающих данных создается единое дерево решений с использованием стандартного подхода. Таким образом, при построении каждого дерева из другого набора данных получается несколько деревьев решений, которые определяются другим набором переменных, узлов и листьев. Наконец, прогнозы этих деревьев объединяются по критерию большинства голосов. Таким образом, мы получаем алгоритм, носящий название *bagging decision trees* [Breiman 1996].

Также существует *bagging regression trees* [Polikar 2006]. Базовые деревья регрессии разбивают набор данных на более мелкие группы, а затем подгоняют простую модель (константу) для каждой подгруппы. К сожалению, модель с одним деревом имеет тенденцию быть крайне нестабильной и плохим предиктором. Тем не менее с помощью начальной загрузки (регрессии) суммирования деревьев эта техника может стать достаточно мощной и эффективной. Кроме того, это обеспечивает фундаментальную основу для более сложных моделей на основе деревьев, таких как случайные леса и машины для повышения градиента [Marin, Vazquez, Lopez, Amores, Leibe 2013].

Существует много методологий построения деревьев регрессии, но одна из самых старых известна как подход дерева классификации и регрессии (CART), разработанный Брейманом (1984). Базовые деревья регрессии разбивают набор данных на более мелкие подгруппы, а затем подбирают простую константу для каждого наблюдения в подгруппе. Разделение достигается последовательными двоичными разделениями (или рекурсивным разделением) на основе разных предикторов. Предсказываемая константа основана на средних значениях отклика для всех наблюдений, которые попадают в эту подгруппу.

Еще одним интересным «бэггинг» алгоритмом является *bagging k-nearest neighbor classifiers*. Зачастую данный алгоритм применяется для распознавания образов. В распознавании образов алгоритм *k*-ближайших соседей (*k*-NN) является непараметри-

ческим методом, используемым для классификации и регрессии [Hall, Samworth 2005, p. 363–379]. В обоих случаях вход состоит из k ближайших обучающих примеров в пространстве признаков. Выходные данные зависят от того, используется ли k -NN для классификации или регрессии.

В классификации k -NN выходные данные являются членами класса. Объект классифицируется большинством голосов своих соседей, при этом объект присваивается классу, наиболее распространенному среди его k ближайших соседей (k является положительным целым числом, обычно небольшим). Если $k = 1$, то объект просто присваивается классу этого единственного ближайшего соседа.

В k -NN-регрессии выводом является значение свойства для объекта. Это значение является средним из значений его k ближайших соседей. k -NN – это тип обучения на основе экземпляров или отложенного обучения, когда функция аппроксимируется только локально, а все вычисления откладываются до классификации. Алгоритм k -NN является одним из самых простых алгоритмов машинного обучения.

Особенность алгоритма k -NN заключается в том, что он чувствителен к локальной структуре данных.

«Стэкинг» алгоритмы

Стэкинг (с англ. – «штабелирование»), или стековое обобщение (по англ. “stacked generalization”) – метод ансамблирования моделей, используемый для объединения информации из нескольких прогнозирующих моделей для создания новой модели. Зачастую полученная модель (модель 2-го уровня) будет превосходить каждую из отдельных моделей благодаря своей сглаживающей природе и способности выделять каждую базовую модель на данных, где она работает лучше всего и дискредитировать на «плохих» данных. По этой причине для стэкинга удобно выбирать существенно различных экспертов [Yu, Luo, Wu, Han, You, Leung, Zhang 2016].

Наиболее простым стэкинг алгоритмом является Straight Stack (также известный как Mean Stack). Этот метод представляет собой простой случай более распространенных методов суммирования, таких как суперстек или усеченный средний стек. Прямой стек суммирует значения амплитуды выборки в местоположениях изохроны и делит на число значений для всех каналов, которые должны быть обработаны:

$$a_t = \frac{1}{N} \sum_{i=1}^N S_i. \quad (4)$$

N представляет собой число значений изохроны, S_i представляет собой амплитуду в месте выборки, а a_t представляет собой амплитуду суммированной трассы в соответствующий момент времени.

Несколько модифицированным «стэкинг» алгоритмом является *stacking with predefined weighting* («стэкинг» алгоритм с заданным весом). Веса обычно подбираются тремя способами: взвешивание путем приглушения (значения амплитуды в каждой выборке в наборе присваиваются с весом 0 или 1), взвешивание функцией смещения (наибольший вес получают выборки с большим смещением) и взвешивание функцией ответа (веса выставляются на основе отношения общего отклика системы к отклику первичных сигналов).

Также определять веса можно, основываясь на изучаемых данных. К таким алгоритмам относится *Optimum Weighted Stack (OWS)* (оптимально взвешенный стек). На основе оптимальных критериев этот алгоритм используется перед суммированием, включая оптимальные фильтры стекирования. Алгоритм для значения суммирования s_t основан на:

$$r_{j,t}, \quad j = 1, 2, \dots, J \text{ и } t = 1, 2, \dots, T$$

$$s_t = \sum_{j=1}^J (w_j r_{j,t}), \quad (5)$$

где $r_{j,t}$ представляет наборы, которые должны быть сложены, w_j – веса, T – количество выборок на набор и J – количество наборов на систему.

Совершенно отличным «стэкинг» алгоритмом является *Iterative Stack, Superstack*. Термин «итеративный стэкинг» часто используется как синоним термина *Superstack*. При итерационных методах особенно важно заботиться об отраженных сигналах с обращением фазы на дальнем смещении. Это требует предварительной обработки или уменьшения до более близкого смещения.

Итеративное суммирование с помощью суперстека основано на разделении амплитуд, положительных a_j и отрицательных b_j , для

всех времен отражения с амплитудой r_j . Число итераций равно n , после первой итерации для матрицы данных выполняется $n = 1$. Коэффициент нормы M для сумм после итерации называется кратностью [Bastos 2014]. В базовой форме алгоритм описывается:

$$\begin{aligned}
 a_j^n &= \begin{cases} r_j & r_j > 0 \\ 0 & r_j \leq 0 \end{cases} \\
 b_j^n &= \begin{cases} 0 & r_j \geq 0 \\ r_j & r_j < 0 \end{cases} \\
 s_+^n &= \frac{1}{M} \sum_{j=1}^J a_j^n \quad \text{и} \quad s_-^n = \frac{1}{M} \sum_{j=1}^J b_j^n
 \end{aligned} \tag{6}$$

с суммами s_+^n и s_-^n — изохрон положительных и отрицательных амплитуд в соответствующее время и скорректированных амплитуд r_j собираются, с

$$r_{j,t}, j = 1, 2, \dots, J \text{ и } t = 1, 2, \dots, T. \tag{7}$$

Существует несколько последствий и условий для геонаучной интерпретации и интегрированных систем, вытекающих из различных описанных алгоритмов суммирования.

Условием прямого стека является некоррелированный шум и амплитуды, сопоставимые по величине по наборам данных. В противном случае оптимальным вариантом является взвешенный стек.

Нормализация результатов с использованием скаляра используется в практике, например, как скаляр мощности для нормализации стека. Это общее использование, вычисления зависят от размеров данных, но они просты в реализации и могут быть распараллелены.

Заключение

Суммируя все, что было приведено выше, об основных типах ансамблевых алгоритмов можно сказать следующее:

Бэггинг. В этом случае часто рассматривают однородных слабых учеников, обучают их параллельно и независимо друг от друга, а затем объединяют их, следуя некоторому детерминированному процессу усреднения. Одним из больших преимуществ бэггинга

является его параллелизм. Поскольку различные модели обучаются независимо друг от друга, при необходимости могут использоваться методы интенсивного распараллеливания [Bühlmann, Yu 2002].

Бустинг. В этом случае часто рассматривают однородных слабых учеников, обучают их последовательно адаптивным способом (слабый ученик зависит от предыдущих) и объединяют их, следуя детерминированной стратегии. Бэггинг использует параллельное обучение базовых классификаторов, то есть бэггинг является улучшающим объединением, а бустинг – улучшающим пересечением.

Стекинг. В этом случае часто учитывают разнородных слабых учеников, изучают их параллельно и объединяют их, обучая метамодель для вывода прогноза, основанного на предсказаниях различных слабых моделей. Особенностью является разделение датасета на несколько частей для обучения методов, следовательно, объем обучающей выборки должен быть велик.

На основе проведенного обзора и приведенных в данной статье сведений об алгоритмах была составлена сравнительная таблица, воспользовавшись которой можно выбрать алгоритм для решения необходимой задачи. В качестве основных были выбраны следующие параметры: объем обучающей выборки, виды экспертов и степень достоверности экспертов.

Таблица

Сравнительная таблица алгоритмов,
построенных на системах «слабых экспертов»

| | Объем обучающей выборки | Виды экспертов | Степень достоверности экспертов |
|--|-------------------------|---|---------------------------------|
| Bagging | | | |
| Bagging decision trees | Выборка среднего объема | гетерогенный ансамбль, природа экспертов сильно отличны | Более 80% |
| Bagging regression trees | | гетерогенный ансамбль, позволяющий части экспертам быть одной природы | Более 60% |
| Bagging k-nearest neighbor classifiers | Выборка большого объема | гетерогенный ансамбль | Более 80% |

Окончание табл.

| | Объем обучающей выборки | Виды экспертов | Степень достоверности экспертов |
|--|-------------------------|--|---------------------------------|
| Boosting | | | |
| AdaBoost | Выборка малого объема | гомогенные ансамбли, в основном применяются нейронные сети | 30–60% |
| AnyBoost LogitBoost LPBoost GentleBoost | | гомогенные ансамбли | 30–80% |
| Stacking | | | |
| Straight Stack | Выборка среднего объема | гомогенные ансамбли | Более 65% |
| Stacking with predefined weighting | | различные эксперты в гомо- и гетерогенных ансамблях | Более 50% |
| Optimum Weighted Stack (OWS) | | различные эксперты в гомо- и гетерогенных ансамблях | |
| Iterativer Stack | | различные эксперты в гомо- и гетерогенных ансамблях | |

Литература

- Bastos 2014 – *Bastos J.A.* Ensemble Predictions of Recovery Rates // Journal of Financial Services Research. 2014. Vol. 6 (2). P. 177–193.
- Breiman 1996 – *Breiman L.* Bagging predictors // Machine Learning. 1996. Vol. 24 (2). P. 123–140.
- Bühlmann, Yu 2002 – *Bühlmann P., Yu B.* Analyzing bagging // Annals of Statistics. 2002. Vol. 30 (4). P. 927–961.
- Hall, Samworth 2005 – *Hall P., Samworth R.J.* Properties of Bagged Nearest Neighbour Classifiers // Journal of the Royal Statistical Society. Series B (Statistical Methodology). 2005. Vol. 67, no. 3. P. 363–379.
- Hawelka, Sitko, Kazakopoulos, Beinat 2017 – *Hawelka B., Sitko I., Kazakopoulos P., Beinat E.* Collective prediction of individual mobility traces for users with short data history // PLoS ONE. 2017. Vol. 12 (1). P. 1–14.
- Khoshgoftaar, Van Hulse, Napolitano 2011 – *Khoshgoftaar T.M., Van Hulse J., Napolitano A.* Comparing boosting and bagging techniques with noisy and imbalanced data // IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans. 2011. Vol. 41 (3). P. 552–568.

- Marin, Vazquez, Lopez, Amores, Leibe 2013 – *Marin J., Vazquez D., Lopez A.M., Amores J., Leibe B.* Random forests of local experts for pedestrian detection // Proceedings of the IEEE International Conference on Computer Vision. New York, NY: IEEE, 2013. P. 2592–2599.
- Polikar 2006 – *Polikar R.* Ensemble based systems in decision making // IEEE Circuits and Systems Magazine. 2006. Vol. 6, issue 3. P. 21–45.
- Posada, Narayanan, Hoffmann, Bertram 2011 – *Posada L.F., Narayanan K.K., Hoffmann F., Bertram T.* Ensemble of experts for robust floor-obstacle segmentation of omnidirectional images for mobile robot visual navigation // Proceedings of the IEEE International Conference on Robotics and Automation. New York, NY: IEEE, 2011. P. 439–444.
- Schapire 1990 – *Schapire R.E.* The strength of weak learnability. Machine Learning. 1990. Vol. 5 (2). P. 197–227.
- Shao, Lavesson, Boeva, Shahzad 2016 – *Shao B., Lavesson N., Boeva V., Shahzad R.K.* A mixture-of-experts approach for gene regulatory network inference // International Journal of Data Mining and Bioinformatics. 2016. Vol. 14 (3). P. 258.
- Wan 2017 – *Wan H.* Gradient Descent Boosting: Convergence and Algorithm. May 12, 2017. URL: <https://courses.grainger.illinois.edu/ece543/sp2017/projects/Haohua%20Wan.pdf> (дата обращения 2 июня 2022).
- Yu, Luo, Wu, Han, You, Leung, Zhang 2016 – *Yu Z., Luo P., Wu S., Han G., You J., Leung H., Zhang J.* Incremental semi-supervised clustering ensemble for high dimensional data clustering // 2016 IEEE 32nd International Conference on Data Engineering – ICDE 2016. New York, NY: IEEE, 2016. P. 1484–1485.
- Zhou 2012 – *Zhou Z.-H.* Ensemble Methods Foundations and Algorithms // Journal of Chemical Information and Modeling. 2012. Vol. 53. P. 233.

References

- Bastos, J.A.* (2014), “Ensemble Predictions of Recovery Rates”, *Journal of Financial Services Research*, vol. 46 (2), pp. 177–193.
- Breiman, L. (1996), “Bagging predictors”, *Machine Learning*, vol. 24 (2), pp. 123–140.
- Bühlmann, P., & Yu, B. (2002), “Analyzing bagging”, *Annals of Statistics*, vol. 30 (4), pp. 927–961.
- Hall, P. and Samworth, R.J. (2005), “Properties of Bagged Nearest Neighbour Classifiers”, *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 3, pp. 363–379.
- Hawelka, B., Sitko, I., Kazakopoulos, P., & Beinat, E. (2017), “Collective prediction of individual mobility traces for users with short data history”, *PLoS ONE*, vol. 12 (1), pp. 1–14.
- Khoshgoftaar, T.M., Van Hulse, J., & Napolitano, A. (2011), “Comparing boosting and bagging techniques with noisy and imbalanced data”, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 41 (3), pp. 552–568.

- Marin, J., Vazquez, D., Lopez, A.M., Amores, J., & Leibe, B. (2013), “Random forests of local experts for pedestrian detection”, *Proceedings of the IEEE International Conference on Computer Visio*, IEEE, New York, NY, USA, pp. 2592–2599.
- Polikar, R. (2006), “Ensemble based systems in decision making”, *IEEE Circuits and Systems Magazine*, vol. 6, issue 3, pp. 21–45.
- Posada, L.F., Narayanan, K.K., Hoffmann, F., & Bertram, T. (2011), “Ensemble of experts for robust floor-obstacle segmentation of omnidirectional images for mobile robot visual navigation”, *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, New York, NY, USA, pp. 439–444.
- Schapire, R.E. (1990), “The strength of weak learnability”, *Machine Learning*, vol. 5 (2), pp. 197–227.
- Shao, B., Lavesson, N., Boeva, V., & Shahzad, R.K. (2016), “A mixture-of-experts approach for gene regulatory network inference”, *International Journal of Data Mining and Bioinformatics*, vol. 14 (3), p. 258.
- Wan, H. (2017), “Gradient Descent Boosting: Convergence and Algorithm”, May 12, 2017, available at: <https://courses.grainger.illinois.edu/ece543/sp2017/projects/Haohua%20Wan.pdf> (Accessed 2 June 2022).
- Yu, Z., Luo, P., Wu, S., Han, G., You, J., Leung, H. and Zhang, J. (2016), “Incremental semi-supervised clustering ensemble for high dimensional data clustering”, *2016 IEEE 32nd International Conference on Data Engineering – ICDE 2016*, IEEE, New York, NY, USA, pp. 1484–1485.
- Zhou, Z.-H. (2012), “Ensemble Methods Foundations and Algorithms”, *Journal of Chemical Information and Modeling*, vol. 53, p. 233.

Информация об авторах

Ксения В. Евдокимова, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; ks.evdokimova@mail.ru

Кирилл Л. Тассов, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; ktassov@policesoft.ru

Information about the authors

Ksenia V. Evdokimova, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Baumanskaya Str., Moscow, Russia, 105005; ks.evdokimova@mail.ru

Kirill L. Tassov, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Baumanskaya Str., Moscow, Russia, 105005; ktassov@policesoft.ru

Математика

УДК 517.5

DOI: 10.28995/2686-679X-2022-2-48-58

О некоторых понятиях и теоремах математического анализа

Аллаберди Г. Галканов

*Государственный гуманитарно-технологический университет,
Орехово-Зуево, Московская обл., Россия, agalkanov@yandex.ru*

Аннотация. В статье изложены результаты, полученные автором в поисках инновационных форм и методов представления учебно-методической информации с тем, чтобы добиться лучшего понимания, усвоения и закрепления читаемого учебного материала студентами, а на выходе – умения применять полученные знания на практике.

На основе аксиомы непрерывности множества действительных чисел даны определения непрерывного множества, разрывного множества со скачком и с пробелом. Приведены примеры.

Применение символики и законов математической логики, а также равносильности необходимого условия существования математического понятия и достаточного условия несуществования этого понятия позволили получить ряд значимых результатов касаясь непрерывности функции, локального экстремума функции и точки перегиба графика функции, теоремы Ролля.

В статье показано существование альтернативы к стандартной форме изложения необходимого условия непрерывности функции в точке и необходимого условия непрерывности функции на множестве. Излагается новый критерий непрерывности функции на множестве, что существенно упрощает исследование функции на непрерывность на множестве. Следствием этого критерия явилось новое определение непрерывности функции на множестве. В качестве примера приведены функции Хевисайда и Дирихле.

Некоторые понятия и теоремы, изложенные в статье, используются автором на занятиях по математическому анализу.

Ключевые слова: теорема, альтернативные доказательства теорем, функция, непрерывность функции, точка локального экстремума функции, критическая точка, стационарная точка, критическая точка возможного перегиба графика функции, стационарная точка возможного перегиба графика функции, точка перегиба

© Галканов А.Г., 2022

Для цитирования: Галканов А.Г. О некоторых понятиях и теоремах математического анализа // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 2. С. 48–58. DOI: 10.28995/2686-679X-2022-2-48-58

About some concepts and theorems in mathematical analysis

Allaberdi G. Galkanov

*State Humanitarian-Technological University, Orekhovo-Zuevo,
Moscow Region, Russia, agalkanov@yandex.ru*

Abstract. The article presents the results obtained by the author in search of innovative forms and methods of presenting educational and methodological information in order to achieve a better understanding, assimilation and consolidation of the educational material read by students, and at the output – the ability to apply the knowledge gained in practice.

Based on the axiom of continuity of the set of real numbers, definitions of a continuous set, a discontinuous set with a jump and with a gap are given. As well as the examples.

The use of symbolism and the laws of mathematical logic, as also the equivalence of the necessary condition for the existence of a mathematical concept and a sufficient condition for the non-existence of this concept, allowed us to obtain a number of significant results with regard to the continuity of the function, the local extremum of the function and the inflection point of the function graph in the Roll theorem.

The article shows the existence of an alternative to the standard form of presentation of the necessary condition for the continuity of a function at a point and the necessary condition for the continuity of a function on a set. A new criterion for the continuity of a function on a set is presented, which greatly simplifies the study of a function for continuity on a set. The consequence of the criterion was a new definition of the continuity of a function on a set. Heaviside and Dirichlet functions are given as an example.

Some concepts and theorems set forth in the article are used by the author in classes on mathematical analysis.

Keywords: theorem, alternative proofs of the theorem, function, continuity of the function, point of local extremum of the function, critical point, stationary point, critical point of possible inflection of the graph of the function, stationary point of possible inflection of the graph of the function, inflection point

For citation: Galkanov, A.G. (2022), “About some concepts and theorems in mathematical analysis”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 2, pp. 48–58, DOI: 10.28995/2686-679X-2022-2-48-58

Введение

Курс математического анализа наряду с алгеброй и геометрией является одним из трех базовых курсов для студентов физико-математического профиля. По математическому анализу написано множество учебников, учебных и учебно-методических пособий [Архипов, Садовничий, Чубариков 1999], [Зорич 2019], [Кудрявцев 1981], [Никольский 2001], [Фихтенгольц 1969], [Пискунов 1978]. Несмотря на это, все еще остаются вопросы методического и отчасти научного характера, требующие переосмысления и даже, быть может, обновления установившихся понятий и теорем. Данная статья посвящена таким понятиям и теоремам.

Пусть T и T' – некоторые теоремы. Теоремы T и T' назовем парными теоремами, если теорема T – необходимое условие существования, а теорема T' – достаточное условие несуществования математического понятия S .

Если T и T' – парные теоремы, то, согласно закону контрапозиции, $T \Leftrightarrow T'$. Поэтому если теорема T' доказывается проще, чем теорема T , то вместо теоремы T достаточно доказать теорему T' .

Метод доказательства теоремы T , использующий доказанную теорему T' , назовем методом парных теорем. В статье пять теорем из математического анализа будут доказаны методом парных теорем.

О непрерывных и разрывных множествах

В [Математическая энциклопедия 1982] понятие непрерывного множества определено как линейно упорядоченное множество, все собственные сечения которого являются дедекиндовыми сечениями. Однако традиционно в учебной литературе по математическому анализу определение понятия непрерывного множества не дается, хотя дается аксиома непрерывности множества \mathbf{R} (см., например, [Архипов, Садовничий, Чубариков 1999], [Зорич 2019], [Кудрявцев 1981], [Никольский 2001], [Фихтенгольц 1969]); также отсутствует определение разрывного множества. Поскольку в статье понятия непрерывного и разрывного множеств будут использованы, дадим их определения.

Пусть A, B – непустые множества.

Определение 1. Если $\forall x \in A \forall y \in B [x < y]$, то A называется множеством, предшествующим множеству B и обозначается $A < B$.

Приведем формулировку аксиомы непрерывности из аксиоматики множества \mathbf{R} [Кудрявцев 1981] в терминах дедекиндоваго

сечения. Пусть непустые множества A, B таковы, что $A < B$ и $A + B = \mathbf{R}$.

Аксиома непрерывности. $\forall x \in A \forall y \in B \exists \xi \in \mathbf{R} [\xi \in [x, y] \vee \xi \in (x; y)]$.

По образцу аксиомы непрерывности определим понятие непрерывного множества G , удовлетворяющего следующим условиям:

- 1) G – бесконечное подмножество множества \mathbf{R} ;
- 2) $\forall x \in G (x \in \mathbf{Q} \cup \mathbf{IR})$, т. е. множество G состоит из рациональных и иррациональных чисел.

Так, например, множество \mathbf{Q} удовлетворяет условию (1), но не удовлетворяет условию (2).

Обозначение множества иррациональных чисел \mathbf{IR} включено в список обозначений.

Пусть непустые множества A, B таковы, что $A < B$ и $A + B = G$.

Определение 2. Если $\forall x \in A \forall y \in B \exists \xi \in G [\xi \in [x, y] \vee \xi \in (x; y)]$, то G называется непрерывным множеством.

Пример 1. По определению 2 докажем непрерывность интервала $G = (a; b)$, где $(a; b) \stackrel{def}{=} \{x: a < x < b, a \in \mathbf{R}, b \in \mathbf{R} (a < b), x \in \mathbf{R}\}$. Пусть $c = (a + b)/2$, $A = (a; c)$, $B = (c; b)$. Тогда $A < B$, $A + B = (a; b)$, $\forall x \in A \forall y \in B \exists \xi \in (a; b) [\xi \in [x, y] \vee \xi \in (x; y)]$, следовательно, по определению 2 интервал $(a; b)$ есть непрерывное множество, где за ξ можно взять число $\xi = (x + y)/2$. В самом деле, пусть $\varepsilon_1, \varepsilon_2$ – сколь угодно малые положительные действительные числа. Полагая $\xi \in [x; y] \vee \xi \in (x; y)$, имеем

$$\xi = \frac{1}{2}(x + y) \leq \frac{1}{2}[(b - \varepsilon_1) - (a + \varepsilon_2)] = \frac{1}{2}(b - a) - \frac{\varepsilon_1 + \varepsilon_2}{2} < \frac{1}{2}(b - a) \in (a; b).$$

Отрицая определение 2, получим определение не непрерывного или разрывного множества со скачком.

Определение 3. Если $\exists a \in A \exists b \in B \forall t \in G [t \notin (a; b)]$, где $A < B$ и $A + B = G$, то G называется разрывным множеством со скачком.

Пример 2. $G = \mathbf{N}$. Пусть $A = \{1, 2, \dots, 10\}$, $B = \{11, 12, \dots\}$, $a = 10 \in A$, $b = 11 \in B$. Тогда $A < B$, $A + B = \mathbf{N}$ и $\forall t \in \mathbf{N} [t \notin (10; 11)]$. Следовательно, по определению 3 \mathbf{N} – разрывное множество со скачком.

Определение 4. Если $\forall x \in A \forall y \in B \exists \alpha \in (x; y) [\alpha \notin G]$, где $A < B$ и $A + B = G$, то G называется разрывным множеством с пробелом.

Пример 3. $G = \mathbf{Q}$. Пусть $A = (-\infty; \sqrt{3}) \cap \mathbf{Q}$, $B = (\sqrt{3}; +\infty) \cap \mathbf{Q}$. Тогда $A < B$, $A + B = \mathbf{Q}$. И пусть $x \in A$, $y \in B$ – любые, $\alpha = \sqrt{3} \in (x; y)$. Так как $\alpha = \sqrt{3} \notin \mathbf{Q}$, то по определению 4 \mathbf{Q} – разрывное множество с пробелом.

Теоремы о непрерывных функциях.

Пусть функция $f: X \rightarrow F$ определена в локальной окрестности точки x_0 $U_\delta(x_0) = (x_0 - \delta; x_0 + \delta) \subset X$, где $0 < \delta < \delta_0$ и δ_0 – сколь угодно малое положительное действительное число.

В математическом анализе необходимое условие непрерывности функции f в точке x_0 сводится к существованию предела функции f в этой точке, т. е. $f \in C(x_0) \Rightarrow \lim_{x \rightarrow x_0} f(x)$. Сформулируем и докажем альтернативную теорему.

Теорема 1 (необходимое условие непрерывности функции в точке). Если функция f непрерывна в точке x_0 , то локальная окрестность $U_\delta(x_0)$ есть непрерывное множество.

Теорема 1' (достаточное условие разрывности функции в точке). Если $U_\delta(x_0)$ – разрывное множество, то f – разрывная функция в точке x_0 .

Доказательство. Так как $U_\delta(x_0)$ – разрывное множество, то оно имеет пробел или скачок, что означает: функция f не определена хотя бы в одной точке множества $U_\delta(x_0)$. Тогда по определению f – разрывная функция в точке x_0 . Теорема 1' доказана. Теорема 1 доказана.

Пусть функция f определена на множестве $X \subseteq \mathbf{R}$.

Теорема 2 (необходимое условие непрерывности функции на множестве). Если f – непрерывная функция на множестве X , то X – непрерывное множество.

Теорема 2' (достаточное условие разрывности функции на множестве). Если X – разрывное множество, то f – разрывная функция на множестве X .

Доказательство. Пусть X – разрывное множество, т. е. область определения функции f разрывное множество. Тогда, согласно определению разрывной функции на множестве, существует, по меньшей мере, одна точка разрыва $x_0 \in X$ функции f , т. е. f – разрывная функция на множестве X . Теорема 2' доказана. Теорема 2 доказана.

Теорема 2 не обратима. Так, например, функция Хевисайда

$$h: \mathbf{R} \rightarrow \{0; 1\}, h(x) = \begin{cases} 0, & \text{если } x < 0, \\ 1, & \text{если } x \geq 0 \end{cases}$$

определена на непрерывном множестве \mathbf{R} , но она разрывная функция.

Теорема 3 (критерий непрерывности функции на множестве). Для непрерывности функции $f: X \rightarrow F$ в своей области определения

X необходимо и достаточно, чтобы множество ее значений F было непрерывным множеством.

Прежде чем перейти к доказательству, приведем определение непрерывности функции на множестве.

$$f \in C(X) \stackrel{\text{def}}{\Leftrightarrow} \forall y \in X \forall \varepsilon > 0 \exists \delta > 0 \forall x \in X [|x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon]. \quad (1)$$

Доказательство. Необходимость. Пусть $f \in C(X)$, т. е. f – непрерывная функция на множестве X . Тогда по теореме 2 X – непрерывное множество и, согласно определению (1), множество $F = \{f(x) : f(x) \in (f(y) - \varepsilon; f(y) + \varepsilon), y \in X\}$ есть интервал на координатной прямой Oy , т. е. непрерывное множество (Пример 1).

Достаточность. Пусть F – непрерывное множество. Тогда область определения X функции f должна быть непрерывным множеством, иначе, согласно теореме 2', функция f была бы разрывной функцией на множестве X , и, как следствие этого, F было бы разрывным множеством. Далее допустим, что $f \notin C(X)$, т. е. f – разрывная функция на множестве X . В таком случае, согласно определению разрывной функции на множестве, область значений F функции f должна иметь пробел или скачок, т. е. F должно быть разрывным множеством. Однако это противоречит условию. Теорема 3 доказана.

Следствие из теоремы 3. На разрывных множествах непрерывные функции не существуют.

Теорема 3 дает основание для нового определения понятия непрерывной функции на множестве.

Определение 5. Если F – непрерывное множество, то функция $f: X \rightarrow F$ называется непрерывной функцией на множестве X .

Так, например, разрывность функции Дирихле $f: \mathbf{R} \rightarrow \{0; 1\}$, где

$$f(x) = \begin{cases} 1, & \text{если } x \in \mathbf{Q}, \\ 0, & \text{если } x \in \mathbf{R} \setminus \mathbf{Q} \end{cases}$$

немедленно следует из определения 5 в силу разрывности множества ее значений. Но доказать разрывность функции Дирихле традиционным способом было бы намного сложнее.

О точках локальных экстремумов и перегибов графика функции

Пусть функция f определена в области $X \subseteq \mathbf{R}$, $x_0 \in X$ – внутренняя точка. Приведем традиционную формулировку теоремы Ферма о необходимых условиях существования точки нестрогого локального экстремума функции.

Теорема 4. Если x_0 – точка нестрогого локального экстремума функции f и функция f дифференцируема в точке x_0 , то производная функции f в точке x_0 равна нулю.

В формулировке теоремы 4 не строгость точки локального экстремума x_0 существенна для прямого доказательства этой теоремы. В случае, если x_0 – точка строгого локального экстремума функции f , то данный способ доказательства не применим. С другой стороны, в теории и практике локального экстремума функций строгий локальный экстремум не менее важен, нежели не строгий. Также отметим, что в теореме 4 отсутствует случай, когда $f \notin D(x_0)$, тем не менее, $x_0 \in L$. Так, например, $f(x) = |x| \notin D(0)$, но $x_0 = 0$ – точка строгого локального минимума данной функции. Желая учесть и этот случай, сформулируем полную версию теоремы Ферма как для точек нестрогого, так и для точек строгого локального экстремума функции и докажем ее.

Теорема 5 (полная версия теоремы Ферма). Если x_0 – точка локального экстремума функции f , то в точке x_0 функция f не дифференцируема или дифференцируема и ее производная в этой точке равна нулю, т. е. $x_0 \in L \Rightarrow (f \notin D(x_0) \vee [f \in D(x_0) \wedge f'(x_0) = 0])$.

Теорема 5 приведена в [Кудрявцев 1981], но без доказательства.

Теорема 5'. $\neg(f \notin D(x_0) \vee [f \in D(x_0) \wedge f'(x_0) = 0]) \Rightarrow x_0 \notin L$.

Доказательство.

$$\begin{aligned} & \neg(f \notin D(x_0) \vee [f \in D(x_0) \wedge f'(x_0) = 0]) \Rightarrow x_0 \notin L \Leftrightarrow \\ & \Leftrightarrow \left[\overbrace{(f \in D(x_0) \wedge f \notin D(x_0))}^{\text{ложно}} \vee (f \in D(x_0) \wedge f'(x_0) \neq 0) \Rightarrow x_0 \notin L \right] \Leftrightarrow \\ & \Leftrightarrow [(f \in D(x_0) \wedge f'(x_0) \neq 0) \Rightarrow (x_0 \notin L)]. \end{aligned}$$

Так как $(f'(x_0) \neq 0) \Leftrightarrow (f'(x_0) < 0 \vee f'(x_0) > 0)$, то x_0 – точка убывания или точка роста функции f соответственно, т. е. не является точкой локального экстремума f . Теорема 5' доказана. Теорема 5 доказана. Полная версия теоремы Ферма доказана.

В учебной литературе по математическому анализу теорема 5 до сих пор остается недоказанной теоремой.

Из полной версии теоремы Ферма следует, что локальный экстремум функции может существовать лишь в тех точках, в которых функция f не дифференцируема или дифференцируема и ее производная равна нулю. Это приводит к необходимости введения следующих понятий.

Определение 6. Если функция f определена в точке x_0 , но не дифференцируема в ней, то x_0 называется критической точкой функции f .

Определение 7. Если функция f дифференцируема в точке x_0 и в этой точке ее производная равна нулю, то x_0 называется стационарной точкой функции f .

Отметим, что до сих пор существуют учебные пособия по математическому анализу, где эти понятия не различаются (см., например, [Пискунов 1978]).

Теорема, аналогичная теореме 5, может быть сформулирована и доказана для точек перегиба графика функции.

Теорема 6. Если x_0 – точка перегиба графика функции f , то в точке x_0 дифференциал второго порядка функции f не существует или существует и ее производная второго порядка $f''(x_0)$ равна нулю, т. е.

$$x_0 \in \Pi \Rightarrow (f \notin D^2(x_0) \vee [f \in D^2(x_0) \wedge f''(x_0) = 0]).$$

В учебной литературе по математическому анализу теорема 6 отсутствует.

Теорема 6'. $\neg(f \notin D^2(x_0) \vee [f \in D^2(x_0) \wedge f''(x_0) = 0]) \Rightarrow x_0 \notin \Pi$.

Доказательство.

$$\begin{aligned} & [\neg(f \notin D^2(x_0) \vee [f \in D^2(x_0) \wedge f''(x_0) = 0]) \Rightarrow x_0 \notin \Pi] \Leftrightarrow \\ & \Leftrightarrow \left[\overbrace{(f \in D^2(x_0) \wedge f \notin D^2(x_0))}^{\text{ложно}} \vee (f \in D^2(x_0) \wedge f''(x_0) \neq 0) \Rightarrow x_0 \notin \Pi \right] \Leftrightarrow \\ & \Leftrightarrow [(f \in D^2(x_0) \wedge f''(x_0) \neq 0) \Rightarrow x_0 \notin \Pi]. \end{aligned}$$

Так как $f''(x_0) \neq 0 \Leftrightarrow f''(x_0) < 0 \vee f''(x_0) > 0$, то x_0 – либо точка выпуклости, либо точка вогнутости графика функции f . Тогда x_0 по определению не может быть точкой перегиба графика функции. Теорема 6' доказана. Теорема 6 доказана.

Из теоремы 6 следует, что перегиб графика функции может существовать лишь в тех точках, в которых либо f'' не существует, либо существует и $f''(x_0) = 0$. Это приводит к необходимости введения следующих понятий.

Определение 8. Если функция f определена в точке x_0 , но $f \notin D^2(x_0)$, то $M(x_0, f(x_0))$ называется критической точкой возможного перегиба графика функции f .

Так, например, $M(0, 0)$ – критическая точка возможного перегиба графика функции $f(x) = \begin{cases} -x^2, & x < 0; \\ x^2, & x \geq 0. \end{cases}$

Определение 9. Если $f \in D^2(x_0)$ и $f''(x_0) = 0$, то $M(x_0, f(x_0))$ называется стационарной точкой возможного перегиба графика функции f .

Так, например, $M(c, 0)$ – стационарная точка возможного перегиба графика функции $f(x) = (x-c)^3$, где $c \in \mathbf{R}$.

И в заключение теорему Ролля докажем методом парных теорем.

Теорема 7 (Мишель Ролль). Если функция f непрерывна на сегменте $[a; b]$, дифференцируема в интервале $(a; b)$ и в точках a, b ее значения равны, то в некоторой точке $c \in (a; b)$ ее производная равна нулю:

$$[f \in C[a; b] \wedge f \in D(a; b) \wedge f(a) = f(b)] \Rightarrow \exists c \in (a; b)[f'(c) = 0].$$

В учебной литературе по математическому анализу доказательство теоремы 7 использует теорему Веерштрасса и теорему Ферма (см., например, [Зорич 2019]) и состоит из двух этапов. Однако теорему Ролля можно доказать и без применения теорем Веерштрасса и Ферма, а именно методом парных теорем.

Теорема 7'. $[f \in C[a; b] \wedge f \in D(a; b) \wedge \forall x \in (a; b)[f'(x) \neq 0]] \Rightarrow [f(a) \neq f(b)]$.

Доказательство. По условию теоремы 7'

$$\forall x \in (a; b)[f'(x) \neq 0] \Leftrightarrow [\forall x \in (a; b)[f'(x) < 0] \vee \forall x \in (a; b)[f'(x) > 0]].$$

Поэтому в интервале $(a; b)$ функция f строго убывает или строго возрастает. Тогда $f(a) \neq f(b)$ в силу непрерывности функции f на сегменте $[a; b]$. Теорема 7' доказана. Теорема 7 доказана.

Выводы

1. Введены понятия парных теорем и метода парных теорем.
2. Методом парных теорем доказано 5 теорем.
3. Дано определение непрерывного множества.
4. Даны определения разрывного множества со скачком и с пробелом.
5. Доказана новая версия необходимого условия непрерывности функции в точке.
6. Доказана новая версия необходимого условия непрерывности функции на множестве.
7. Сформулирован и доказан новый критерий непрерывности функции на множестве.
8. Дано новое определение непрерывности функции на множестве.
9. Доказана полная версия теоремы Ферма о необходимом условии существования точки локального экстремума функции.
10. Сформулировано и доказано необходимое условие существования точки перегиба графика функции.
11. Введены понятия критической точки и стационарной точки возможного перегиба графика функции.
12. Предложено новое доказательство теоремы Ролля.

Обозначения

- $\mathbf{N}, \mathbf{Q}, \mathbf{IR}, \mathbf{R}$ – множество натуральных, рациональных, иррациональных и действительных чисел соответственно;
- L – множество точек локального экстремума функции;
- Π – множество точек перегиба графика функции;
- $C(x_0)$ – множество функций, непрерывных в точке $x_0 \in \mathbf{R}$;
- $C(X)$ – множество функций, непрерывных на множестве $X \subseteq \mathbf{R}$;
- $D(x_0)$ – множество функций, дифференцируемых в точке $x_0 \in \mathbf{R}$;
- $D(X)$ – множество функций, дифференцируемых на множестве $X \subseteq \mathbf{R}$;
- $D^2(x_0)$ – множество функций, дважды дифференцируемых в точке $x_0 \in \mathbf{R}$.

Литература

- Архипов, Садовичий, Чубариков 1999 – *Архипов Г.И., Садовичий В.А., Чубариков В.Н.* Лекции по математическому анализу. М.: Высш. школа, 1999. 696 с.
- Зорич 2019 – *Зорич В.А.* Математический анализ. Ч. 1. М.: МЦНМО, 2019. 564 с.

- Кудрявцев 1981 – *Кудрявцев Л.Д.* Математический анализ. Т. 1. М.: Высш. школа, 1981. 496 с.
- Никольский 2001 – *Никольский С.М.* Курс математического анализа. М.: Наука, 2001. 468 с.
- Пискунов 1978 – *Пискунов Н.С.* Дифференциальное и интегральное исчисления. М.: Наука, 1978. 561 с.
- Фихтенгольц 1969 – *Фихтенгольц Г.М.* Курс дифференциального и интегрального исчисления. Т. 1. М.: Наука, 1969. 609 с.

References

- Arkhipov, G.I., Sadovichii, V.A. and Chubarikov, V.N. (1999), *Lektsii po matematicheskoi analizu* [Lectures in Mathematical Analysis], Vyssh. Shkola, Moscow, Russia, 696 p.
- Fikhtengol'ts, G.M. (1969), *Kurs differentsial'nogo i integral'nogo ischisleniya, T. I* [A Course in Differential and Integral Calculus. Vol. I], Nauka, Moscow, Russia, 609 p.
- Kudryavtsev, L.D. (1981), *Matematicheskii analiz, T. I* [Mathematical Analysis. Vol. I], Vyssh. Shkola, Moscow, Russia, 496 p.
- Nikol'skii, S.M. (2001), *Kurs matematicheskogo analiza* [Course in Mathematical Analysis], Nauka, Moscow, Russia, 468 p.
- Piskunov, N.S. (1978), *Differentsial'noe i integral'noe ischisleniye* [Differential and integral calculus], Nauka, Moscow, Russia, 561 p.
- Zorich V.A. (2019), *Matematicheskii analiz* [Mathematical analysis], Part I, MTsNMO, Moscow, Russia, 564 p.

Информация об авторе

Аллаберди Г. Галканов, кандидат технических наук, доцент, Государственный гуманитарно-технологический университет, Орехово-Зуево, Московская обл., Россия; 142600, Россия, Московская обл., Орехово-Зуево, ул. Зеленая, д. 22; agalkanov@yandex.ru

Information about the author

Allaberdi G. Galkanov, Cand. of Sci. (Technical Engineering), associate professor, State Humanitarian-Technological University, Orekhovo-Zuevo, Moscow Region, Russia; bld. 22, Zelenaya Str., Moscow Region, Orekhovo-Zuevo, Russia, 142600; agalkanov@yandex.ru

MMFlow: масштабируемая потоковая модель для генерации изображений

Кирилл Е. Казаков

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
kazakovke@student.bmstu.ru*

Максим А. Кудрявцев

*Московский государственный технический
университет им. Н.Э. Баумана, Москва, Россия,
kudryavtsev@bmstu.ru*

Аннотация. Моделирование естественных изображений является актуальной проблемой в области компьютерного зрения. Генеративные потоковые модели (flow-based models), использующие оценку максимального правдоподобия для изучения распределения многомерных векторов, предоставляют гибкую и масштабируемую архитектуру для описания изображений. Такие модели имеют высокую вычислительную эффективность, а также используют внутреннее представление изображений, которое позволяет изменять конкретные признаки генерируемых данных. Однако потоковые модели, как правило, значительно уступают в качестве получаемых изображений, в сравнении с современными авторегрессионными моделями. В статье предложена вероятностная потоковая модель, основная идея которой предполагает последовательное использование комбинации различных пар бинарных масок на разных уровнях пространственного разрешения. Нововведение способствует улучшению поиска как локальных, так и глобальных признаков изображений. Предлагаемая архитектура легко масштабируется, что позволяет применять модель для генерации изображений различных размеров, в том числе изображений высокого разрешения, без потери стабильности обучения. Работоспособность модели продемонстрирована на четырех наборах изображений различного разрешения, достигнуты значительные улучшения на стандартных метриках оценки качества сгенерированных изображений, сокращен разрыв в эффективности между потоковыми и авторегрессионными моделями.

Ключевые слова: генерация изображений, глубокое обучение, потоковые модели, обратимые преобразования

© Казаков К.Е., Кудрявцев М.А., 2022

Для цитирования: Казаков К.Е., Кудрявцев М.А. MMFlow: масштабируемая потоковая модель для генерации изображений // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 2. С. 59–76. DOI: 10.28995/2686-679X-2022-2-59-76

MMFlow. Scalable flow-based model for image generation

Kirill E. Kazakov

*Bauman Moscow State Technical University, Moscow, Russia,
kazakovke@student.bmstu.ru*

Maksim A. Kudryavtsev

*Bauman Moscow State Technical University, Moscow, Russia,
kudryavtsev@bmstu.ru*

Abstract. Modeling of natural images is an urgent issue in the field of computer vision. Generative flow-based models, which use maximum likelihood estimation to learn the distribution of multidimensional vectors, provide a flexible and scalable architecture for image description. Such models have high computational efficiency, and also use an internal representation of images that allows changing specific features of the generated data. However, flow-based models, as a rule, have much worse quality of the generated images in comparison with modern autoregressive models. The article proposes a probabilistic flow-based model, the main idea of which involves the consistent use of a combination of different pairs of binary masks at multiple levels of spatial resolution. The innovation improves the search for both local and global features of images. The proposed architecture is easily scaled, which allows the model to be used to generate images of various sizes, including high-resolution images, without loss of learning stability. The model's performance was demonstrated by four datasets of images with different resolutions, significant improvements were achieved on standard metrics for evaluating the quality of generated images, reducing the efficiency gap between flow-based and autoregressive models.

Keywords: image generation, deep learning, flow-based models, invertible transformations

For citation: Kazakov, K.E. and Kudryavtsev, M.A (2022), "MMFlow. Scalable flow-based model for image generation", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, no. 2, pp. 59–76, DOI: 10.28995/2686-679X-2022-2-59-76

Введение

Проблема получения массивов данных о реальных объектах, например изображений, видео- и аудиосигналов, стоит крайне остро, так как во многих областях машинного обучения для создания и валидации моделей требуются большие объемы аннотированных данных. Стоимость получения таких данных может быть высокой. Поэтому генеративные модели, используемые для создания новых выборок, представляют собой быстро развивающуюся область исследований, которая охватывает широкий спектр задач, таких как генерация и стилизация изображений.

Генерация распределений изображений с высоким разрешением дополнительно усложняет создание качественных моделей, поскольку подобные данные имеют, как правило, значительно более сложную структуру, а также большее количество взаимосвязей, деталей и признаков, которые должны быть определены моделью. Также следует учитывать, что память графического процессора зачастую является ограничивающим фактором для современных архитектур глубоких нейронных сетей. Потребность в памяти для обучения модели существенно возрастает с увеличением числа ее параметров, вследствие чего снижается максимально доступный объем пакета обрабатываемых во время обучения данных, что усугубляет проблему качества обучения.

Обзор подходов к генерации изображений

В настоящее время для генерации изображений наиболее широко используются авторегрессионные модели [Ak, Xu, Lin et al. 2020], генеративно-состязательные сети [Goodfellow, Pouget-Abadie, Mirza et al. 2014] и потоковые модели [Tomczak 2022]. Ниже коротко рассмотрены наиболее популярные представители каждой группы.

Авторегрессионные модели

Представители этой группы, например, PixelRNN [Oord, Kalchbrenner and Kavukcuoglu 2016], рассматривают изображение как последовательность пикселей и представляют его вероятность через произведение условных вероятностей всех пикселей. При этом вероятность интенсивности каждого пикселя зависит от ин-

тенсивностей всех ранее сгенерированных пикселей. Такой подход позволяет эффективно обобщать данные и создавать четкие изображения, однако генерация изображений сопряжена с высокими вычислительными затратами. Модели этого типа не имеют скрытого представления, поскольку непосредственно моделируют условное распределение пикселей, что ограничивает сферу их применимости.

Генеративно- состязательные сети

Для генерации изображений используется их скрытое представление. Сети этой группы состоят из двух нейронных сетей, одна из которых обучена генерировать данные, а другая – отличать созданные изображения от реальных. Несмотря на наличие проблем со стабильностью процесса обучения и вариативностью получаемых данных, генеративно-состязательные сети создают четкие изображения и могут быть использованы для генерации изображений высокого разрешения.

Потоковые модели

Модели этой группы формируют вектор входных данных из пространства, вероятностное распределение которого заранее известно. С помощью формулы замены переменной последовательность обратимых преобразований используется для получения выходного изображения. Для обучения используется оценка максимального правдоподобия. Преимуществом потоковых моделей, таких как Real NVP [Dinh, Sohl-Dickstein and Bengio 2016], является высокая скорость генерации изображений, однако показатели качества генерации существенно ниже, чем у генеративно-состязательных или авторегрессионных моделей.

Потоковые модели представляют собой относительно новую перспективную группу нейронных сетей. В рамках статьи предложено улучшение модели Real NVP, повышающее стабильность обучения, качество генерируемых изображений, решающее проблему высокой зависимости модели от наборов данных для обучения.

Общий подход к построению потоковых моделей

Основная идея потоковых моделей заключается в применении формулы замены переменных для моделирования распределения наблюдаемых изображений p_x с помощью скрытого многомерного пространства Z с заданным априорным распределением p_z . Как правило, для p_z выбирается нормальное распределение. Для этого используется биекция $f: X \rightarrow Z$, поиск которой и является основной задачей модели:

$$p_x(x) = p_z(z) \left| \det \left(\frac{\partial g(z)}{\partial z^T} \right) \right|,$$

где x – пространство изображений,

z – скрытое пространство,

$g = f^{-1}$ – обратное преобразование,

$\frac{\partial g(z)}{\partial z^T}$ – якобиан функции g в точке z .

Якобиан функции g в точке z определяется следующим образом:

$$\det \left(\frac{\partial g(z)}{\partial z^T} \right) = \det \begin{pmatrix} \frac{\partial g_1}{\partial z_1}(z) & \dots & \frac{\partial g_1}{\partial z_n}(z) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial z_1}(z) & \dots & \frac{\partial g_n}{\partial z_n}(z) \end{pmatrix}$$

где n – размерность пространства Z .

Таким образом, для получения элемента исходного распределения $x \sim p_x$ необходимо выбрать элемент $z \sim p_z$, а затем выполнить преобразование $x = f^{-1}(z) = g(z)$.

Для эффективного использования замены переменной выбирается особый вид преобразования f , который позволяет получить треугольную матрицу якобиана, тем самым существенно упростив его вычисление. Функция представляет собой последовательность простых биекций, так называемых аффинных соединительных слоев, в каждой из которых одна часть входного вектора остается неизменной, а над второй выполняется обратимое преобразование, зависящее от первой части. Пусть D – размер

входного вектора x , $d < D$, тогда соединительный слой может быть представлен следующим образом:

$$\begin{cases} y_{1:d} = x_{1:d} \\ y_{d+1:D} = x_{d+1:D} \otimes \exp(s(x_{1:d}) + t(x_{1:d})), \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{1:d} = y_{1:d} \\ x_{d+1:D} = (y_{d+1:D} - t(x_{1:d})) \otimes \exp(-s(y_{1:d})), \end{cases}$$

где \otimes – поэлементное умножение, $s, t : R^d \mapsto R^{D-d}$.

Обратное преобразование не требует поиска обратных к s и t функций (рис. 1), что позволяет применять произвольные преобразования. Так, для модели Real NVP использовались глубокие сверточные нейросети.

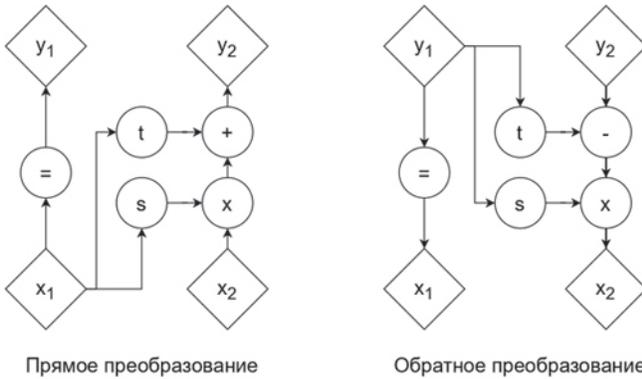


Рис. 1. Графики прямого и обратного преобразований

Система 2 может быть записана с использованием бинарной маски b :

$$y = b \otimes x + (1 - b) \otimes (x \otimes \exp(s(b \otimes x) + t(b \otimes x))).$$

Использование композиции слоев с различными масками позволяет устранить недостаток соединительного слоя, который после преобразования оставляет неизменной часть входного вектора. При этом вычисление обратного преобразования композиции $f_a \circ f_b$ может быть выполнено по формуле:

$$(f_a \circ f_b)^{-1} = f_b^{-1} \circ f_a^{-1}$$

Многомасштабная архитектура [Alexander 2005] применяется для использования соединительных слоев на различных пространственных разрешениях изображения. К изображению размера $s \times s \times c$ (s – ширина и высота, c – количество каналов) применяется часть соединительных слоев, после чего оно преобразуется в изображение меньшего масштаба, но с большим числом каналов: $\frac{s}{2} \times \frac{s}{2} \times 4c$, затем процесс преобразования и масштабирования продолжается. Во избежание громоздких вычислений в связи с большим числом слоев, при изменении масштаба выполняется прореживание переменных модели [Simonyan and Zisserman 2015]: одна половина подвергается дальнейшим преобразованиям, а вторая добавляется непосредственно к итоговому распределению скрытого пространства Z (рис. 2). Как следствие, в модели определяются промежуточные уровни представления, соответствующие признакам изображения различных размеров. Модификация позволяет распределить функцию потерь по нейронной сети, а также существенно снижает необходимый объем вычислений и памяти.

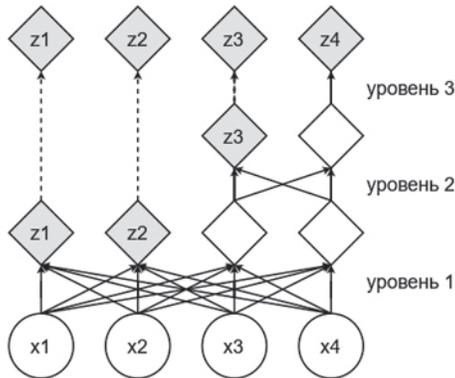


Рис. 2. Прореживание переменных

Улучшенная композиция соединительных слоев MMFlow

В предлагаемой модели MMFlow рекурсивно производится изменение масштаба изображения до тех пор, пока выход последнего слоя не станет вектором размера $c \times 4 \times 4$. Например, для исходного

изображения размера $3 \times 64 \times 64$ будет использовано пять масштабных уровней, а для размера $3 \times 256 \times 256$ – семь.

Бинарные маски в соединительных слоях применяются попарно – за каждым слоем с определенной маской следует слой с тем же типом маски, но инвертированным значением. На каждом уровне используется несколько различных пар масок, причем для слоев с большим пространственным разрешением уместно использовать большее количество масок.

После обработки изображения на последнем соединительном слое уровня следует увеличение масштаба изображения и прореживание. Таким образом, осуществляется переход на новый масштабный уровень половины выходных измерений текущего уровня. На рис. 3 представлена конфигурация уровней и масок модели MMFlow на примере изображений размера 64×64 . Используемые коды масок соединительных слоев приведены на рис. 6. Серым цветом обозначена инвертированная бинарная маска.

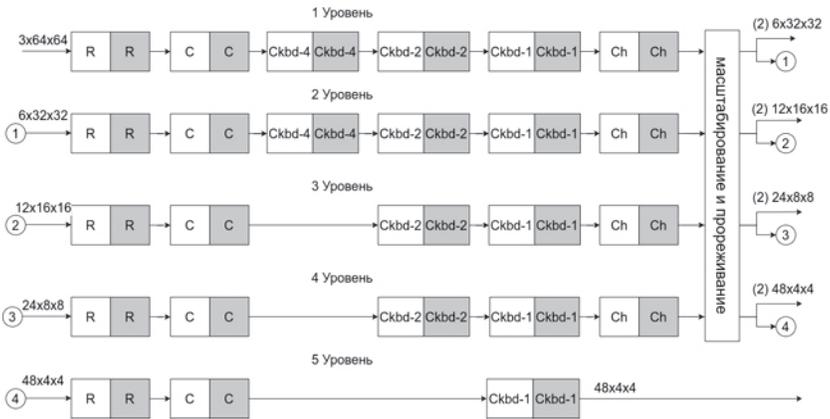


Рис. 3. Конфигурация модели MMFlow

В качестве функций преобразования s и t внутри каждого соединительного слоя используются сверточные нейросети с пропускными соединениями. Идея пропускных соединений заключается в добавлении исходного значения к выходному, полученному обработкой этого значения в остаточном блоке, состоящем из нескольких сверточных слоев. Пропуск входным сигналом части блоков сети позволяет решить проблему затухающих градиентов, и тем самым предоставляет возможность

обучения глубоких моделей с большим числом слоев [He, Zhang, Ren et al. 2016].

Для описания каждого соединительного слоя используется нейросеть с шестью сверточными слоями и двумя пропускными соединениями, на вход которой подаются объединенные векторы параметров функций s и t , сеть одновременно применяет преобразования s и t к соответствующим частям входного вектора. На рис. 4 показана конфигурация сети, в скобках указан размер ядра свертки. Первый и последний сверточные слои блока используются для изменения числа каналов входного и выходного векторов соответственно, внутренние слои используют одинаковое число фильтров, которое зависит от пространственного разрешения входных данных и, следовательно, отличается на разных масштабных уровнях.

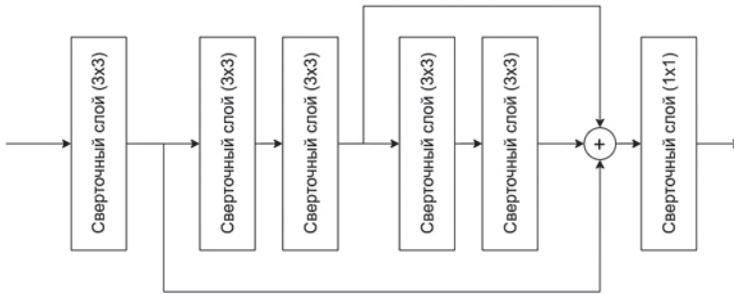


Рис. 4. Схема остаточной сети, используемой в соединительных слоях

Использование сверточных слоев позволяет учесть двумерную топологию изображения. Основная идея заключается в применении свертки двух матриц – входного изображения и фильтра (ядра), который представляет из себя матрицу весов. Ядро скользит по высоте и ширине входного изображения и в каждой пространственной позиции вычисляется поэлементное произведение ядра и соответствующей области изображения, в ячейку выходного вектора заносится сумма полученных произведений. При этом каждый фильтр имеет столько же каналов, сколько и входное изображение. Выходным значением является вектор, число каналов которого определяется количеством фильтров, а пространственный размер – размером фильтра. На выходной размер также могут влиять другие параметры, такие как шаг

фильтра (пропуск части позиций изображения при движении ядра) и дополнение отступа (расширение изображения, например дополнение нулями по краям). Последний шаг в сверточном слое – добавление к выходному значению вектора смещения. На рис. 5 приведена схема свертки трехканального изображения 6×6 с двумя фильтрами с шагом 1.

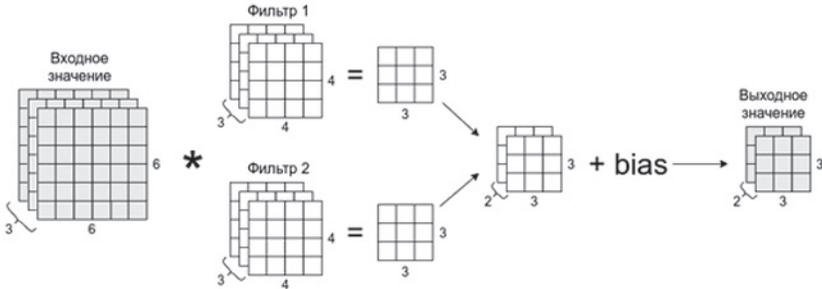


Рис. 5. Свертка изображения с использованием нескольких фильтров

В модели Real NVP используется всего два вида масок: шахматная доска с шагом 1, представленная на рис. 6(а), а также маскировка по каналам – рис. 6(б). При этом на одном масштабном уровне несколько раз применяются соединительные слои с одинаковой конфигурацией масок, что препятствует поиску важных признаков изображения как в пространственном, так и в канальном измерениях. Для решения описанной проблемы были разработаны две новые маски – чередование строк, представленное на рис. 6(д), и чередование столбцов изображения – рис. 6(г), а также была расширена идея маски шахматной доски путем изменения размера клеток – рис. 6(в): если в исходной модели использовалась шахматная маска с размером клетки 1, то для разработанной модели были также использованы маски с размерами клетки 2 и 4. Примеры масок, включая как новые, так и использованные ранее, приведены на рис. 6. В квадратных скобках приведены коды, соответствующие маскам. Полученные маски позволяют на каждом масштабном уровне оценивать не только локальные взаимосвязи между данными, как в случае шахматной маски с единичной клеткой, но и изучать более широкие области, а также общие паттерны всего изображения.



Рис. 6. Схемы бинарных масок для соединительных слоев

На каждом уровне вначале используются маски разбиения по строкам и столбцам, затем применяется одна или несколько пар слоев с шахматными масками разных размеров клетки. Для выбора количества вариантов шахматной маски в зависимости от пространственного разрешения предлагается использовать следующее правило: применять шахматные маски с размерами клетки, равными степеням двойки, от большего размера к меньшему. Максимальный размер клетки на каждом уровне выбирается из расчета, что на последнем уровне (с пространственным разрешением 4×4) используется только единичная клетка, а с увеличением разрешения каждые 2 уровня добавляется новый размер клетки до достижения выбранного предельного значения. Завершающей на каждом уровне (за исключением последнего) является пара слоев с разбиением по каналам.

В качестве функции активации нейросети используется ReLU. Для повышения стабильности обучения также использовались пакетная и весовая нормализации.

При обучении на изображениях с малым разрешением (32 и 64) количество фильтров во внутренних сверточных слоях равнялось 16 на первом масштабном уровне и увеличивалось вдвое на каждом последующем. Применялись пакеты размером по 32 изображения. Для работы с изображениями большего разрешения использовался только один остаточный блок с начальным количеством фильтров, равным 8, размер пакета также был установлен 8.

Для обучения сети использовался оптимизатор Adam [Kingma and Ba 2014] со скоростью обучения 10^{-3} и добавлением L^2 регуляризации с коэффициентом $5 \cdot 10^{-5}$.

Полученные результаты

Для проверки эффективности модели MMFlow было произведено обучение на нескольких, наиболее популярных наборах данных,

которые обычно используются для оценки глубоких генеративных моделей. Использованные наборы изображений представлены в табл. 1. Изображения каждого набора приводились к единому разрешению, указанному в таблице.

Таблица 1

Наборы данных

| Название набора изображений | Разрешение изображений | Количество изображений | Время обучения (число эпох) |
|---|------------------------|------------------------|-----------------------------|
| CIFAR-10 [Krizhevsky 2012] | 32×32 | 60 тыс. | 400 |
| ImageNet [Russakovsky, Deng, Su et al. 2015] | 64×64 | 1,2 млн | 40 |
| Large-scale Scene-Understanding (LSUN). Church outdoor. [Yu, Zhang, Song et al. 2015] | 64×64 | 126 тыс. | 100 |
| Oxford 102 Flowers [Nilsback and Zisserman 2008] | 256×256 | 6 тыс. | 300 |

В модели Real NVP предлагалось использование числа уровней, необходимого для сокращения пространственного размера изображения до 4×4 на выходном слое, при этом на каждом уровне использовалась одинаковая конфигурация масок, состоящая из повторяющихся шахматных и канальных масок. Данный вариант архитектуры использовался для обучения модели Real NVP на наборе данных Oxford 102 Flowers размера 256×256 , таким образом применялось семь уровней масштабирования.

В табл. 2 приведены результаты сравнения моделей в количестве бит на измерение (bpd – bits per dim). Каждый цветовой компонент пикселя изображения считается отдельным измерением, т. е. RGB-изображение размера 32×32 имеет $32 \times 32 \times 3 = 3072$ измерения. Несмотря на отставание от авторегрессионных моделей, таких как PixelRNN, которое характерно для потоковых моделей, MMFlow существенно улучшает показатели на каждом наборе данных относительно исходного варианта модели, а также успешно конкурирует с другими потоковыми моделями.

Таблица 2

Результаты генерации изображений в единицах bpd

| Модель | CIFAR-10 (32×32) | ImageNet (64×64) | LSUN church (64×64) | Oxford 102 Flowers (256×256) |
|---|---------------------|---------------------|------------------------|------------------------------------|
| MMFlow | 3.31 | 3.84 | 2.82 | 2.46 |
| Real NVP | 3.49 | 3.98 | 3.08 | 2.72 |
| Glow [Kingma and Dhariwal 2018] | 3.35 | 3.81 | – | – |
| Flow++ [Ho, Chen, Srinivas et al 2019] | 3.08 | 3.69 | – | – |
| PixelRNN | 3.00 | 3.63 | – | – |

Было установлено, что модель MMFlow также более устойчива при обучении на изображениях высокого разрешения, чем Real NVP. На рис. 7 представлены кривые обучения для набора данных Oxford 102 Flowers на протяжении 300 эпох. Помимо различия в значении, к которому сходятся модели, с увеличением числа эпох также становится заметен разброс значений модели Real NVP, включая довольно значительные выбросы, достигающие значений 0.2 bpd.

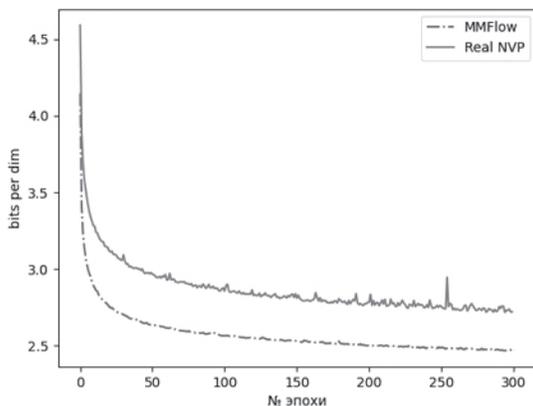


Рис. 7. Кривые обучения моделей MMFlow и Real NVP на наборе данных Oxford 102

На рис. 8 представлено сравнение образцов изображений набора данных и сгенерированных моделью MMFlow. Сгенерированные изображения выглядят глобально согласованными. Часть изображений имеет локальные области с достаточно низкой резкостью, но данный эффект становится менее заметен с увеличением разрешения. Для наборов ImageNet и LSUN модель отражает наличие у изображения переднего и заднего плана, общие концепции освещения – яркость и согласованность теней с направлением источника света.

Наборы данных, показанные на рисунке, расположены в следующем порядке: CIFAR-10, ImageNet (64), LSUN (church, 64):



Рис. 8. Левый столбец – оригинальные изображения, правый – результат генерации модели MMFlow

На рис. 9 продемонстрированы результаты генерации изображений с разрешением 256×256 для двух моделей – Real NVP (первый ряд) и MMFlow (второй ряд). Изображения, сгенерированные моделью MMFlow, выглядят более резкими, а также лучше отражают общую структуру изображения – форму цветка и тональное разделение переднего и заднего планов.



Рис. 9. Примеры изображений, сгенерированных моделями Real NVP и MMFlow

Заключение

В статье предложена модель для генерации изображений, модифицирующая архитектуру на основе потоков путем использования новых бинарных масок и их комбинаций для соединительных слоев на разных уровнях разрешения. Результаты экспериментов демонстрируют преимущество разработанного метода над базовым подходом модели Real NVP как с точки зрения визуальной оценки качества получаемых изображений, так и на основе сравнения оценок правдоподобия.

Модель MMFlow предоставляет возможность масштабирования путем увеличения числа уровней разрешения и добавления новых соединительных слоев для обучения на изображениях с разрешением до 256×256 без необходимости применения явных многомасштабных подходов или решений по искусственному увеличению размера выходных изображений [Wang, Chen and Hoi 2020], что подтверждается устойчивостью сети при обучении на изображениях большого размера.

Литература

Ak, Xu, Lin, et al. 2020 – Ak, K., Xu, N., Lin, Z. et al. Incorporating Reinforced Adversarial Learning in Autoregressive Image Generation // Computer Vision – ECCV 2020 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part 21. New York: Springer, 2020.

- Alexander 2005 – *Alexander S.* Multiscale Methods in Image Modelling and Image Processing. Ontario: University of Waterloo, 2005.
- Devroye 1986 – *Devroye L.* Sample-based non-uniform random variate generation. New York, NY: Springer-Verlag, 1986.
- Dinh, Sohl-Dickstein, Bengio 2016 – *Dinh L., Sohl-Dickstein J., Bengio S.* Density estimation using Real NVP // 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017. Toronto: ICLR, 2016.
- Goodfellow, Pouget-Abadie, Mirza, et al. 2014 – *Goodfellow I.J., Pouget-Abadie J., Mirza M. Et al.* Generative Adversarial Nets // 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014. Red Hook, NY: Curran Associates, 2014.
- He, Zhang, Ren et al. 2016 – *He K., Zhang X., Ren S. Et al.* Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. New York, NY: IEEE, 2016. P. 770–778.
- Ho, Chen, Srinivas, et al 2019 – *Ho J., Chen X., Srinivas A. et al.* Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design // 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019. Red Hook, NY: Curran Associates, 2019. P. 2722–2730.
- Kingma, Ba 2014 – *Kingma D., Ba J.* Adam: A Method for Stochastic Optimization // 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015. Red Hook, NY: Curran Associates, 2015.
- Kingma, Dhariwal 2018 – *Kingma D., Dhariwal P.* Glow: Generative Flow with Invertible 1×1 Convolutions // 32nd International Conference on Neural Information Processing Systems, Montreal, Canada, 3–8 December 2018. Red Hook, NY: Curran Associates, 2018. P. 10236–10245.
- Krizhevsky 2012 – *Krizhevsky A.* Learning Multiple Layers of Features from Tiny Images // Technical Report TR-2009. Toronto: University of Toronto, 2012.
- Nilsback, Zisserman 2008 – *Nilsback M.-E., Zisserman A.* Automated Flower Classification over a Large Number of Classes // *Sixth Indian Conference on Computer Vision, Graphics & Image*, Jodhpur, India, 19–22 December 2021. Red Hook, NY: Curran Associates, 2008. P. 722–729.
- Oord, Kalchbrenner, Kavukcuoglu 2016 – *Oord A., Kalchbrenner N., Kavukcuoglu K.* Pixel Recurrent Neural Networks // 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016. Red Hook, NY: Curran Associates, 2016. P. 1747–1756.
- Russakovsky, Deng, Su, et al. 2015 – *Russakovsky O., Deng J., Su H. et al.* ImageNet Large Scale Visual Recognition Challenge // *International Journal of Computer Vision*. 2015. Vol. 115, no. 3. P. 211–252.
- Simonyan, Zisserman 2015 – *Simonyan K., Zisserman A.* Very Deep Convolutional Networks for Large-Scale Image Recognition // 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015. Toronto: ICLR, 2015.

- Tomczak 2022 – Tomczak J. Flow-Based Models // Deep Generative Modeling. Cham: Springer, 2022. P. 27–56.
- Wang, Chen, Hoi 2020 – Wang Z., Chen J., Hoi S. Deep Learning for Image Super-Resolution: A Survey // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. Vol. 43, no. 10. P. 3365–3387.
- Yu, Zhang, Song et al. 2015 – Yu F., Zhang Y., Song S. Et al. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. URL: <https://arxiv.org/pdf/1506.03365v3.pdf> (дата обращения 5 июня 2022).

References

- Ak, K., Xu, N., Lin, Z. et al. (2020), “Incorporating Reinforced Adversarial Learning in Autoregressive Image Generation”, *16th European Conference on Computer Vision*, 23–28 August 2020, Springer, New York, USA.
- Alexander, S. (2005), *Multiscale Methods in Image Modelling and Image Processing*, University of Waterloo, Ontario, Canada.
- Devroye, L. (1986), *Sample-based non-uniform random variate generation*, Springer-Verlag, New York, NY, USA.
- Dinh, L., Sohl-Dickstein, J. and Bengio, S. (2016), “Density estimation using Real NVP”, *5th International Conference on Learning Representations*, Toulon, France, 24–26 April 2017, ICLR, Toronto, Canada.
- Goodfellow, I.J., Pouget-Abadie, J., Mirza, M. et al. (2014), “Generative Adversarial Nets”, *28th International Conference on Neural Information Processing Systems*, Montreal, Canada, 8–13 December 2014, Curran Associates, Red Hook, NY, USA, 2014.
- He, K., Zhang, X., Ren, S. et al. (2016), “Deep Residual Learning for Image Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016, Curran Associates, Red Hook, NY, USA, pp. 770–778.
- Ho, J., Chen X., Srinivas A. et al. (2019), “Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design”, *36th International Conference on Machine Learning*, New York City, NY, USA, 9–15 June 2019, Curran Associates, Red Hook, NY, USA, pp. 2722–2730.
- Kingma, D. and Ba, J. (2014), “Adam: A Method for Stochastic Optimization”, *3rd International Conference on Learning Representations*, San Diego, CA, USA, 7–9 May 2015, Curran Associates, Red Hook, NY, USA.
- Kingma, D. and Dhariwal, P. (2018), “Glow: Generative Flow with Invertible 1×1 Convolutions”, *32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, 3–8 December 2018, Curran Associates, Red Hook, NY, USA, pp. 10236–10245.
- Krizhevsky, A. (2012), “Learning Multiple Layers of Features from Tiny Images”, *Technical Report TR-2009*, University of Toronto, Toronto, Canada.
- Nilsback, M.-E. and Zisserman, A. (2008), “Automated Flower Classification over a Large Number of Classes”, *Sixth Indian Conference on Computer Vision, Graphics &*

- Image*, Jodhpur, India, 19–22 December 2021, Curran Associates, Red Hook, NY, USA, pp. 722–729.
- Oord, A., Kalchbrenner, N. and Kavukcuoglu, K. (2016), “Pixel Recurrent Neural Networks”, *33rd International Conference on Machine Learning*, New York City, NY, USA, 19–24 June 2016, Curran Associates, Red Hook, NY, USA, pp. 1747–1756.
- Russakovsky, O., Deng, J., Su, H. et al. (2015), “ImageNet Large Scale Visual Recognition Challenge”, *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252.
- Simonyan, K. and Zisserman, A. (2015), “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *3rd International Conference on Learning Representations*, San Diego, CA, USA, 7–9 May 2015, ICLR, Toronto, Canada.
- Tomczak, J. (2022), “Flow-Based Models”, *Deep Generative Modeling*, Springer, Cham, Switzerland, pp. 27–56.
- Wang, Z., Chen, J. and Hoi, S. (2020), “Deep Learning for Image Super-Resolution: A Survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, October 2021, pp. 3365–3387.
- Yu, F., Zhang, Y., Song, S. et al. (2015), “LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop”, available at: <https://arxiv.org/pdf/1506.03365v3.pdf> (Acceded 5 June 2022).

Информация об авторах

Кирилл Е. Казаков, студент, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 107005, Россия, Москва, 2-я Бауманская ул., д. 5, kazakovke@student.bmstu.ru

Максим А. Кудрявцев, аспирант, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 107005, Россия, Москва, 2-я Бауманская ул., д. 5, kudryavtsev@bmstu.ru

Information about the authors

Kirill E. Kazakov, student, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Bauman Str., Moscow, Russia, 107005; kazakovke@student.bmstu.ru

Maksim A. Kudryavtsev, postgraduate student, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Bauman Str., Moscow, Russia, 107005; kudryavtsev@bmstu.ru

Дизайн обложки

Е.В. Амосова

Корректор

Ж.П. Григорьева

Компьютерная верстка

Н.В. Москвина

Подписано в печать 28.06.2022.

Формат $60 \times 90^{1/16}$.

Уч.-изд. л. 4,5. Усл. печ. л. 4,9.

Тираж 1050 экз. Заказ № 1619

Издательский центр
Российского государственного
гуманитарного университета
125047, Москва, Миусская пл., 6
www.rsuh.ru