THE MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION



ISSN 2687-0517

Computing, Telecommunications and Control

Vol. 15, No. 4 2022

Peter the Great St. Petersburg Polytechnic University 2022

COMPUTING, TELECOMMUNICATIONS AND CONTROL

EDITORIAL COUNCIL

Prof. Dr. Rafael M. Yusupov corresponding member of RAS, St. Petersburg Institute for Informatics and Automation of the RAS, Russia,

Prof. Dr. Sergey M. Abramov corresponding member of RAS, full member of RAS, Ailamazyan Program Systems Institute of the RAS,

Prof. Dr. Dmitry G. Arseniev corresponding member of RAS, Peter the Great St. Petersburg Polytechnic University, Russia,

Prof. Dr. Vladimir V. Voevodin corresponding member of RAS, Lomonosov Moscow State University, Russia,

Prof. Dr. Vladimir S. Zaborovsky, Peter the Great St. Petersburg Polytechnic University, Russia,

Prof. Dr. Vladimir N. Kozlov, Peter the Great St. Petersburg Polytechnic University, Russia,

Prof. Dr. Alexandr E. Fotiadi, Peter the Great St. Petersburg Polytechnic University, Russia,

Prof. Dr. Igor G. Chernorutsky, Peter the Great St. Petersburg Polytechnic University, Russia.

EDITORIAL BOARD

Editor-in-chief

Prof. Dr. Alexander S. Korotkov, Peter the Great St. Petersburg Polytechnic University, Russia;

Members:

Assoc. Prof. Dr. Vladimir M. Itsykson, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. Philippe Ferrari, Grenoble Alpes University, France;

Prof. Dr. Yevgeni Koucheryavy, Tampere University of Technology, Finland;

Prof. Dr. Wolfgang Krautschneider, Hamburg University of Technology, Germany;

Prof. Dr. Fa-Long Luo, University of Washington, USA;

Prof. Dr. Sergey B. Makarov, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. Emil Novakov, Grenoble Alpes University, France;

Prof. Dr. Nikolay N. Prokopenko, Don State Technical University, Russia;

Prof. Dr. Mikhail G. Putrya, National Research University of Electronic Technology, Russia;

Sen. Assoc. Prof. Dr. Evgeny Pyshkin, University of Aizu, Japan;

Prof. Dr. Viacheslav P. Shkodyrev, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. Peter V. Trifonov, ITMO University, Russia;

Prof. Dr. Igor A. Tsikin, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. Sergey M. Ustinov, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. Lev V. Utkin, Peter the Great St. Petersburg Polytechnic University, Russia.

Open access journal is to publish articles of a high scientific level covering advanced experience, research results, theoretical and practical problems of informatics, electronics, telecommunications, and control.

The journal is indexed by Ulrich's Periodicals Directory, Google Scholar, EBSCO, ProQuest, Index Copernicus, VINITI RAS Abstract Journal (Referativnyi Zhurnal), VINITI RAS Scientific and Technical Literature Collection, Russian Science Citation Index (RSCI) database Scientific Electronic Library and Math-Net.ru databases.

The journal is registered with the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR). Certificate $\Im J No. \Phi C77-77378$ issued 25.12.2019.

Editorial office

Dr. Sc., Professor A.S. Korotkov - Editor-in-Chief;

E.A. Kalinina – literary editor, proofreader; G.A. Pyshkina – editorial manager; A.A. Kononova – computer layout; D.Yu. Alekseeva – English translation. Address: 195251 Polytekhnicheskaya Str. 29, St. Petersburg, Russia.

+7 (812) 552-6216, e-mail: infocom@spbstu.ru

Release date: 20.01.2023

© Peter the Great St. Petersburg Polytechnic University, 2022

The journal is included in the List of Leading PeerReviewed Scientific Journals and other editions to publish major findings of PhD theses for the research degrees of Doctor of Sciences and Candidate of Sciences.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ



ISSN 2687-0517

Информатика, телекоммуникации и управление

Том 15, № 4 2022

Санкт-Петербургский политехнический университет Петра Великого 2022

ИНФОРМАТИКА, ТЕЛЕКОММУНИКАЦИИ И УПРАВЛЕНИЕ

РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

Юсупов Р.М., чл.-кор. РАН, Санкт-Петербургский институт информатики и автоматизации РАН, Санкт-Петербург, Россия; *Абрамов С.М.*, чл.-кор. РАН, Институт программных систем им. А.К. Айламазяна РАН, Москва, Россия; *Арсеньев Д.Г.*, чл.-кор. РАН, д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Воеводин В.В.*, чл.-кор. РАН, Московский государственный университет им. М.В. Ломоносова, Москва, Россия; *Заборовский В.С.*, д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Козлов В.Н.*, д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Фотшади А.Э.*, д-р физ.-мат. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Черноруцкий И.Г.*, д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Черноруцкий И.Г.*, д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия; *Черноруцкий И.Г.*, д-р техн. наук,

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА

Главный редактор

Коротков А.С., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Редакционная коллегия:

Ицыксон В.М., канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Феррари Ф., профессор, Университет Гренобль-Альпы, Гренобль, Франция;

Краутинайдер В., профессор, Гамбургский технический университет, Гамбург, Германия;

Кучерявый Е.А., канд. техн. наук, профессор, Университет Тампере, Финляндия.

 $\Pi \omega \phi$.- Π ., University of Washington, Washington, USA;

Макаров С.Б., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Новаков Э., профессор, Университет Гренобль-Альпы, Гренобль, Франция;

Прокопенко Н.Н., д-р техн. наук, профессор, Донской государственный технический университет, г. Ростовна-Дону, Россия;

Путря М.Г., д-р техн. наук, профессор, Национальный исследовательский университет «Московский институт электронной техники», Москва, Россия;

Пышкин Е.В., профессор, Университет Айзу, Айзу-Вакаматсу, Япония;

Трифонов П.В., д-р техн. наук, доцент, Национальный исследовательский университет ИТМО, Санкт-Петербург, Россия;

Устинов С.М., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Уткин Л.В., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Цикин И.А., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия;

Шкодырев В.П., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия.

Сетевое издание открытого доступа публикует статьи высокого научного уровня, освещающие передовой опыт, результаты НИР, теоретические и практические проблемы информатики, электроники, телекоммуникаций, управления.

Сведения о публикациях представлены в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich`s Periodical Directory», в Российской государственной библиотеке. В базах данных: Российский индекс научного цитирования (РИНЦ), Google Scholar, EBSCO, Math-Net.Ru, ProQuest, Index Copernicus.

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ЭЛ № ФС77-77378 от 25.12.2019.

Учредитель и издатель: Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация. Редакция журнала

д-р техн. наук, профессор А.С. Коротков – главный редактор;

Е.А. Калинина – литературный редактор, корректор; Г.А. Пышкина – ответственный секретарь, выпускающий редактор;

А.А. Кононова – компьютерная вёрстка; Д.Ю. Алексеева – перевод на английский язык.

Адрес редакции: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.

Тел. редакции +7(812) 552-62-16, e-mail: infocom@spbstu.ru

Дата выхода: 20.01.2023

© Санкт-Петербургский политехнический университет Петра Великого, 2022

Журнал с 2002 года входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Contents

Intellectual Systems and Technologies

Prokofiev K.V., Ivanov S.V. SAR images generation from optical imagery					
Zhu Y. Formation of flight control for a group of unmanned aerial vehicles based on algorithm of multi-agent swarm model					
Vlasenko N.A., Dusaeva A.I., Nikiforov I.V., Prelovskii D.S. Technique for automating charging of an electric vehicle based on a Raspberry Pi controller using neural networks	37				
Antonov A.P., Besedin D.S., Filippov A.S. Research and comparative analysis of the effectiveness of software and hardware implementations of the operation of summing transposed matrices	51				
The Seventh Conference on Software Engineering and Information Management (SEIM-2022)					
Pismenny A.A., Sokolov E.A. Token-wise approach to span-based question answering	64				
Fadeeva E.S., Ershov V.A. Multi-channel transformer: A transformer-based model for multi-speaker speech recognition	73				
Tarasov D.A., Ershov V.A. Fixing 1-bit Adam and 1-bit LAMB algorithms	86				
Kobzarenko D.N., Mustafaev A.G., Gasanova Z.A., Magomedova D.S. One-dimensional convolutional layers in a neural network for wind speed time series analysis	98				

Содержание

Интеллектуальные системы и технологии

Прокофьев К.В., Иванов С.В. Генерация радиолокационных снимков местности из оптических 7					
 нжу Ю. Формирование управления полетом группы беспилотных летательных аппаратов на основе алгоритма многоагентной модели роения					
Власенко Н.А., Дусаева А.И., Никифоров И.В., Преловский Д.С. Методика автоматизации зарядки электромобиля на базе котроллера Raspberry Pi с использованием нейронных сетей.	37				
Антонов А.П., Беседин Д.С., Филиппов А.С. Исследование и сравнительный анализ эффективности программной и аппаратных реализаций операции суммирования транспонированных матриц	51				
Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)					
Письменный А.А., Соколов Е.А. Потокенный подход к поиску диапазона правильного ответа в вопросно-ответных системах	64				
Фадеева Е.С., Ершов В.А. Многоканальный трансформер: Модель для распознавания многоголосной речи, основанная на архитектуре трансформер					
Тарасов Д.А., Ершов В.А. Решение проблем алгоритмов 1-bit Adam и 1-bit LAMB	86				
Кобзаренко Д.Н., Мустафаев А.Г., Гасанова З.А., Магомедова Д.С. Одномерные сверточные слои в нейронной сети для анализа временных рядов – скоростей ветра					

Интеллектуальные системы и технологии Intellectual Systems and Technologies

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15401 УДК 004.932



ГЕНЕРАЦИЯ РАДИОЛОКАЦИОННЫХ СНИМКОВ МЕСТНОСТИ ИЗ ОПТИЧЕСКИХ

К.В. Прокофьев¹ ⊠, С.В. Иванов²

^{1,2} Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО), Санкт-Петербург, Российская Федерация

^{III} baterflyrity@yandex.ru

Аннотация. Рост технологического уровня промышленности за последние годы привёл к широкому применению технологий статистического вывода, например, нейронных сетей и машинного обучения, требующих наличия большого объёма исходных данных. В области обработки данных с беспилотных летательных аппаратов наблюдаются сложности с получением исходных радиолокационных снимков местности в связи с дороговизной оборудования, организацией бесполётных зон и трудностью согласований сбора данных. Наборы данных в свободном доступе ограничены и нецелесообразны для ряда исследований. Исследовательские группы либо работают в условиях отсутствия данных, либо нанимают геодезистов и 3D-дизайнеров. В статье изучены методы решения этой проблемы путём генерации радиолокационных изображений из общедоступных оптических спутниковых карт, которые доступны для любой точки Земли. Рассмотрены два метода: моделирование физических процессов распространения радиоволн и свёртка цветного изображения в градации серого. Физическая модель строится в первом приближении как линейное распространение электромагнитной волны с одной точкой отражения, несущественные эффекты распространения радиоволн в атмосфере и при отражении опускаются. Описаны различные алгоритмы свёртки, линейного преобразования цветового пространства RGB в градации серого: YPbPr, HSV, линейная регрессия. Физическая модель отбрасывается ввиду практической неприменимости и сложности реализации. После оценки результатов свёртки по критериям визуального и математического максимального правдоподобия предпочтение отдаётся алгоритму YPbPr. Предложены дополнительные шаги для более точной генерации радиолокационных снимков: добавление шума и преобразования пространства. Полученный алгоритм генерирует визуально и математически адекватные псевдорадиолокационные снимки. Результаты применяются для создания наборов данных, сопоставленных радиолокационных и оптических изображений. Метод имеет преимущество перед аналогами на сельском и диком клаттерах, но проигрывает в густонаселённых районах. Данные используются в нейронных сетях и машинном обучении. Предложено последующее комбинирование алгоритма с генерацией точечных объектов интереса.

Ключевые слова: радиолокационные изображения, генератор, SAR, преобразование спутниковых карт, обработка аэросъёмки

Для цитирования: Прокофьев К.В., Иванов С.В. Генерация радиолокационных снимков местности из оптических // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 7–21. DOI: 10.18721/JCSTCS.15401

Intellectual Systems and Technologies

Research article DOI: https://doi.org/10.18721/JCSTCS.15401 UDC 004.932



SAR IMAGES GENERATION FROM OPTICAL IMAGERY

K.V. Prokofiev¹ \square , S.V. Ivanov²

^{1,2} ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics), St. Petersburg, Russian Federation Baterflyrity@yandex.ru

Abstract. Technological level boost in the industry has led to widespread utilization of neural networks and machine learning that require large datasets. Data processing in unmanned aerial vehicles faces several difficulties such as high equipment cost, no-fly areas and complicated imagery permission acquisition. Public datasets can be deficient and impractical. Researches are forced to either work without initial data or hire surveyors and 3D-designers. The paper solves this problem by generation of synthetic aperture radar images from publicly available optical satellite maps available anywhere on Earth. Two methods are discovered: physical processes of radio wave propagation modeling and color image convolution into grayscale. The physical model is constructed in the first approximation as a linear propagation of an electromagnetic wave with a single point of reflection, insignificant radio waves atmosphere propagation and reflection effects are omitted. Various algorithms of convolution, linear transformation of RGB color space into grayscale are examined: YPbPr, HSV, linear regression. The physical model is discarded due to its practical inapplicability and complexity of implementation. After evaluating the convolution results according to the maximum likelihood criteria, preference is given to the YPbPr algorithm. Additional steps are proposed for more accurate generation: noise addition and space transformation. The resulting algorithm generates visually and mathematically adequate pseudo-radar images to obtain initial datasets, correlated radar and optical images. The datasets are supposed to improve neural networks. The method gives an advantage over analogues in rural and wild clutters but loses in urban areas. In further studies, we propose to combine the results with objects of interest generation.

Keywords: radar imagery, generator, SAR, satellite map transformation, optic to SAR translation, aerial imagery processing

Citation: Prokofiev K.V., Ivanov S.V. SAR images generation from optical imagery. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 7–21. DOI: 10.18721/JCSTCS.15401

Введение

Бурное развитие технологий в направлении машинного обучения и искусственного интеллекта способствует их обширному прикладному применению в различных областях промышленности, в частности, в секторе беспилотных летательных аппаратов (БПЛА).

Собираемые в процессе полётов данные подвергаются широкому спектру разнообразных видов анализа, в первую очередь с помощью нейронных сетевых технологий, которые требуют значительного объёма априорных данных для обучения и оптимизации. Сбор данных БПЛА затруднён из-за целого ряда причин: дороговизны оборудования [1, 2], организации бесполётных зон над точками интереса и крупными населёнными пунктами [3–5], сложностью и медленной скоростью согласования полётов БПЛА [6], ограниченным доступом к собираемым данным. Исследователям и аналитикам приходится заимствовать чужие наборы данных, которые в некоторых видах работ являются не релевантными или получены на другой местности. Сбор исходных данных для целевой местности нередко является невозможной задачей. Альтернативой приходится генерация необходимых данных.



Рис. 1. Примеры оптического (слева) и радиолокационного (справа) снимка земной поверхности сельской местности Fig. 1. Examples of optical (left) and radar (right) images of the Earth's surface of rural areas

Традиционными классами собираемых данных являются показания датчиков, геодезические координаты, оптические снимки и радиолокационные снимки (рис. 1). Последний класс разительно отличается сложностью съёмки в связи с более высокой стоимостью используемого оборудования и другими приведёнными выше факторами. На рисунке видно, что запечатлённая поверхность включает в себя поле, луг, асфальтовую автодорогу, отдельные деревья и их группы, часть стадиона, кустарную аллею.

Радиолокационные изображения — аналогичные оптическим изображения, полученные с использованием электромагнитных волн отличных частот из диапазона радио [7, 8]. Такие изображения обладают рядом преимуществ перед оптическими аналогами [9]: не зависят от метеоусловий и освещённости, являются более качественными и детальными, просвечивают оптический камуфляж. Но требуют использования специального дорогого оборудования. Снимок состоит из нескольких слоёв: рельеф местности, подстилающая поверхность, объекты интереса. Классическим направлением в генерации является симуляция отражения радиоволн от объектов интереса и последующим вычислением радиолокационной картины объекта в том числе с помощью нейронных сетей. Данному направлению посвящено множество работ [10, 11]. Направление допускает слабую корреляцию и зависимость результата анализа данных от окружающего объект интереса фона. Действительно, для анализа наиболее часто используются свёрточные нейронные сети, которые выделяют на радиолокационном снимке область, включающую объект интереса, и работают далее с ней. Подобный тип анализа данных с БПЛА является лишь одним из применимых; в исследовательских задачах требуется полная радиолокационная картина местности.

Рассматривается второе менее популярное направление, генерация клаттерной карты (карты типов подстилающей поверхности и строений), объединяющей в себе рельеф местности с подстилающей поверхностью, на которую при необходимости возможно нанести требуемые объекты интереса, описанные выше, без ощутимых потерь качества результирующего снимка. Особенностью данного направления является его малая степень изученности, что выражается в слабой представленности тематических практических работ в открытом доступе [12], и отсутствие исходных наборов данных для применения нейронных сетей в открытом доступе.



Рис. 2. Пример спутникового радиолокационного снимка целевой местности (рис. 1) из общедоступного набора данных Shuttle Radar Topography Mission
 Fig. 2. An example of a satellite radar image of the target area (Fig. 1) from the public data set of the Shuttle Radar Topography Mission

Метод SARViz [13] позволяет осуществлять симуляцию радиосъёмки на основании заранее подготовленной трёхмерной модели целевой местности. В случае отсутствия модели предлагается использовать мировые радиолокационные спутниковые снимки Shuttle Radar Topography Mission [14], которые имеют малое разрешение, что приводит к низкому качеству симуляции (рис. 2). Разрешение снимка не позволяет работать в целевых масштабах. Методы RaySAR [15, 16] и CohRaS [17] применяют упрощённую физическую модель отслеживания отражений радиоволн для получения детальных радиоснимков исходных моделей местности и моделей объектов интереса. Система [18] задействует комплексный подход: дополнительно к использованию методов отслеживания отражений радиоволн интегрирует в себя метод генерации трёхмерных моделей произвольной земной поверхности на основе открытых данных геоинформационных систем, полагая их априорное наличие в достойном качестве.

Задействование заранее подготовленных моделей ограничивает применение высококачественной симуляции преимущественно городской местностью. Моделирование сельской, малоизученной, труднодоступной местности не является целесообразным для ряда задач, в том числе оперативного развёртывания систем поддержки принятия решений на основе снимков местности.

Широкий спектр работ посвящён обратной задаче — преобразованию радиолокационных снимков в оптические [19, 20].

Исследовательские группы вынуждены работать либо в условиях отсутствия исходных данных, либо нанимать компании для геодезического обследования и последующего моделирования местности. Такой подход является неприемлемым по разным причинам: в первую очередь ввиду ограниченности финансирования. Традиционно данная проблема решается путём подмены условий эксплуатации конечного продукта хорошо изученными данными схожей местности или полным отказом от использования радиолокации.

В связи с развитием технологий машинного обучения и искусственного интеллекта задействование и наращивание подобных допущений всё больше приближается к критической точке. В статье исследуется метод решения изложенной проблемы путём применения общедоступных общемировых оптических спутниковых снимков местности в генеративном алгоритме сопутствующих радиолокационных снимков.

Физическое моделирование

Физическая радиолокационная съёмка в первом приближении моделируется линейным распространением радиоволны с одной точкой отражения на земной поверхности (см. рис. 3; под спутником подразумевается любой БПЛА); объёмность пространства и некоторые эффекты распространения электромагнитной волны в атмосфере не учитываются. Радиоволны мощностью Рtx излучаются в сторону Земли в момент времени, когда БПЛА находится в левой позиции. Затем часть энергии радиоволны отражается (волна испытывает потери L_r) от поверхности Земли, приобретая свойства, характеризующие поверхность отражения (клаттера). За время отражения и возвращения волны «наверх» БПЛА перемещается в правую позицию, где детектирует отражённую волну мощностью P_{r_x} и извлекает из неё переносимую информацию о клаттере или объекте интереса. В процессе распространения в атмосфере радиоволны теряют часть энергии (испытывают потери d).

Модель можно записать логарифмическим уравнением радиосвязи:

$$P_{R_x} = P_{T_x} + G_{T_x} - L_{T_x} - L_{los} - L_R - L_{los} - L_{R_x} + G_{R_x},$$
(1)

где T_x – передача; R_x – приём; P – мощность; L – потери на оборудовании; L_{los} – потери в свободном пространстве; L_R – потери при отражении от поверхности Земли.

Потери в свободном пространстве вычисляются по следующей формуле [21]:

$$L_{los} = 32, 4 + 20 \lg(f) + 20 \lg(d),$$
⁽²⁾

где f – частота радиоволны, МГц; d – длина радиотрассы – расстояние от БПЛА до точки отражения, км.

Неровность поверхности Земли определяется в виде разброса Δd (рис. 4). Влияние разброса вычисляется по следующей формуле:

$$\Delta L_{los} = L_{los} \left(d + \Delta d \right) - L_{los} \left(d \right) = 20 \lg \left(1 + \frac{\Delta d}{d} \right). \tag{3}$$

Допуская, что высота полёта летательного аппарата составляет 1 км, разрешающая способность изображения 1 м, угол отклонения линии распространения радиоволны от горизонта принимается за 90 градусов, а длина радиотрассы d оценивается в 1 км. Тогда среднестатистический разброс Δd при относительно ровной земной поверхности в сельской, лесной, полевой и пригородной местности [22] составляет 15 м. Влияние разброса ΔL_{los} равняется 0,129 дБ или 3 % и не оказывает значительного влияния на радиолокационную картину. Уравнение радиосвязи можно переписать к виду

$$P_{R_r} = C - L_R \tag{4}$$

и перенормировать к виду

$$Y = K_R, (5)$$



Puc. 3. Принцип радиолокационной съёмки в первом приближении Fig. 3. The principle of radar survey in the first approximation



Рис. 4. Неровность поверхности Земли Fig. 4. Unevenness of the Earth's surface

где C – константа; Y – яркость изображения; K_R – коэффициент отражения радиоволны заданной частоты от поверхности Земли.

Радиолокационное изображение формируется под действием различной отражающей способности Земли в условиях заданной местности. Потери при отражении зависят от клаттера или объекта интереса и могут быть выведены из данных геоинформационных систем. Точность модели определяется в первую очередь точностью геоинформационных систем, что не подходит для решения общей задачи, а только лишь для генерации набора данных на хорошо исследованной местности, и приносит мало практической пользы. Решено отказаться от физического моделирования и исследовать метод максимального правдоподобия путём преобразования цветового пространства оптических снимков местности RGB в яркостное.

Преобразование цветового пространства

Извлечение радиосвойств клаттеров и объектов интереса из оптического снимка принципиально некорректно и невозможно с точки зрения физики. Но применение сгенерированных псевдорадиолокационных в нейронных сетях, машинном обучении и других алгоритмах обработки данных с БПЛА целесообразно. Рассматривается задача подбора, попиксельная свёртка цветовых компонент оптического снимка в оттенок серого. Задача преобразования цветовых пространств в оттенки серого и оптимизации яркостной компоненты *Y*, в общем случае, не имеет физических или математических ограничений. Однако исследуется исключительно линейная свёртка, различные широко практикуемые алгоритмы: YPbPr, HSV, линейная регрессия.

Яркостная компонента цветовой модели YPbPr вычисляется по формуле

$$Y = K_R R + \left(1 - K_R - K_B\right) G + K_B B, \tag{6}$$

где R, G, B – цветовые компоненты пространства RGB от 0 до 1; K – коэффициенты гамма-коррекции. Коэффициенты гамма-коррекции для телевидения стандартной чёткости определены как $K_{R} = 0,144, K_{R} = 0,299$ [23].

Яркостная компонента цветовой модели HSV вычисляется по формуле

$$Y = \max\left(R; \; G; \; B\right). \tag{7}$$

Коэффициенты a линейной регрессии оценены методом наименьших квадратов с нулевым свободным членом, полученные значения равняются -0,156 для R; 2,189 для G; -1,713 для B:

$$Y = a_R R + a_G G + a_R B. \tag{8}$$

Отрицательное значение коэффициента для цветовой компоненты *R* выглядит странно с учётом физической основы цветовой модели *RGB*, но допустимо в условиях абстрагирования от физических принципов и решения сугубо оптимизационной задачи.

Все рассмотренные алгоритмы генерации исследуются на наборе сопоставленных оптических и радиолокационных снимков местности. После генерации псевдорадиолокационного снимка производится нормализация его яркости: добавляется свободный член, обеспечивающий минимальное расхождение средней яркости эталонных радиолокационных и сгенерированных псевдорадиолокационных снимков.

Оценка результатов

Оценка рассмотренных алгоритмов генерации псевдорадиолокационных изображений из оптических вычисляется следующим образом:

$$\overline{E} = \frac{\sum_{W} \sum_{H} \left| Y - \hat{Y} \right|}{WH} \tag{9}$$

в абсолютной величине \overline{E} – средняя ошибка на пиксель (принимает значения от 0 до 256), где W – ширина изображения, H – высота изображения, Y – истинная яркость, \hat{Y} – сгенерированная яркость.

В качестве дополнительных метрик используются максимальная (10) и минимальная (11) ошибка на пиксель:

$$E_{\max} = \max_{W, H} \left| Y - \hat{Y} \right|,\tag{10}$$

$$E_{\min} = \min_{W,H} \left| Y - \hat{Y} \right|. \tag{11}$$

Произведена оценка и сравнение (табл. 1) сгенерированных снимков (рис. 5), в том числе с эталонными радиолокационными (рис. 1, справа). По результатам предлагается к использованию алгоритм YPbPr (6).

Оптимизация алгоритма

После генерации изображения в градациях серого рекомендуется дополнительно наложение шума для симуляции помех, поглощающих элементов и неоднородностей. По природе шум является мультипликативным, по характеру распределения подчиняется гамма-закону [24, 25].



Рис. 5. Сгенерированные псевдорадиолокационные изображения (алгоритмы слева направо: YPbPr, HSV, линейная регрессия) Fig. 5. Generated pseudo radar images (algorithms from left to right: YPbPr, HSV, linear regression)

Таблица 1

Оценка результатов генерации псевдорадиолокационных снимков

Table 1

Алгоритм	\overline{E}	E_{\min}	E _{max}	Среднее смещение яркости	Визуальная оценка
YPbPr	25	0	212	+33	Наиболее визуально схожее изображение
HSV	26	0	214	+37	Слишком высветленные, однородные поля и луга
Линейная регрессия	21	0	200	-6	Наиболее визуально расхожее изображение

Evaluation of the results of pseudo radar images generation

На наборе исходных данных проведена оценка параметров распределения мультипликативного шума (рис. 6) методом максимального правдоподобия [26—29] на окрестности радиуса 3 пикселя. Графики плотности вероятности наглядно демонстрируют сходимость гамма распределения к более простому, нормальному, согласно центральной предельной теореме:

$$\overline{Y}_{[x,y]}(R) = (2R+1)^{-2} \sum_{\Delta x = -R}^{R} \sum_{\Delta y = -R}^{R} Y_{[x+\Delta x, y+\Delta y]},$$
(12)

где $\overline{Y}_{[x,y]}$ – среднее значение окрестности пикселя по координатам (x; y); R – радиус окрестности; $Y_{[x,y]}$ – значение пикселя по координатам (x; y).

Рекомендуется наложение мультипликативного гауссовского шума с параметрами $\mu = 0.9885796517360985$, $\sigma = 0.2508281649583726$ (рис. 7).

После генерации радиолокационного изображения рекомендуется выполнить симуляцию полёта беспилотного летательного аппарата (рис. 8, 9). Данный порядок действий позволяет со-



Puc. 6. Сравнение распределения мультипликативного шума Fig. 6. Comparison of multiplicative noise distribution



Рис. 7. Пример наложения нормального мультипликативного шума на полученное псевдорадиолокационное изображение с применением алгоритма YPbPr

Fig. 7. An example of superimposing normal multiplicative noise on the obtained pseudo radar image using the YPbPr algorithm

кратить в три раза сложность итоговой генерации. Всё многообразие аэродинамических явлений [30–32] в первом приближении сводится к модели преобразований пространства, которая включает:

- 1. Наклон.
- 2. Аффинные преобразования.
- 2.1. Растяжение (методом обратного сжатия).



Рис. 8. Пример исходного оптического изображения Fig. 8. An example of the original optical image



Рис. 9. Примеры полученных из исходного (рис. 8) псевдорадиолокационных изображений с наложением шума и применением преобразований пространства Fig. 9. Examples of pseudo radar images obtained from the original (Fig. 8) with noise overlay and the use of space transformations

- 2.2. Движение.
- 2.2.1. Параллельный перенос.
- 2.2.2. Поворот.

Под растяжением методом обратного сжатия подразумевается предварительное сжатие изображения (перенормировка масштаба), чтобы при растяжении не приходилось применять интерполяцию пикселей.

Параметры преобразований подбираются случайным образом и составляют не более 10 % от изображения для предотвращения функциональных искажений.

Совокупный набор параметров (табл. 2) позволяет получить конечные псевдорадиолокационные изображения из близких по масштабу к исследованным оптических снимков местности из открытых источников данных.

Таблица 2

Параметры генерации псевдорадиолокационных снимков

Table 2

Параметр	Значение
Коэффициент гамма-коррекции синего канала	0,144
Коэффициент гамма-коррекции красного канала	0,299
Смещение яркости	+ 33
Математическое ожидание нормального распределения шума	0,9885796517360985
Среднеквадратическое отклонение нормального распределения шума	0,2508281649583726
Параметры преобразований пространства	до 10 %

Parameters for generating pseudo radar images

Заключение

Разработан алгоритм генерации псевдорадиолокационных снимков земной поверхности на основе общедоступных оптических изображений, таких как онлайн-карты. Предложены четыре этапа генерации: вычисление яркостной компоненты цветового пространства YPbPr, смещение яркостной компоненты, добавление нормального мультипликативного шума, преобразования пространства (наклон, растяжение методом обратного сжатия, параллельный перенос, поворот). Произведена генерация тестовых изображений, выполнена математическая и визуальная оценка результатов. Получаемые псевдорадиолокационные изображения применимы для обучения нейронных сетей и других алгоритмов анализа данных, собираемых БПЛА. Разработанный метод позволяет ускорить, удешевить или детализировать исследования за счёт использования общедоступных исходных данных для любой локации по сравнению с аналогичными методами на сельском и диком клаттерах, но проигрывает в качестве на густонаселённых и хорошо изученных районах. Рекомендуется комбинация предложенного алгоритма для генерации радиолокационного снимка местности с рассмотренными методами генерации объектов интереса.

СПИСОК ЛИТЕРАТУРЫ

1. Keating E.G., Kerman J., Arthur D. Usage patterns and costs of unmanned aerial systems. Congressional budget office. 2021 // URL: http://www.cbo.gov/publication/57090

2. Svedin J., Bernland A., Gustafsson A., Claar E., Luong J. Small UAV-based SAR system using low-cost radar, position, and attitude sensors with onboard imaging capability // Internat. J. of Microwave and Wireless Technologies. 2021. vol. 13 (6). Pp. 602–613. DOI: 10.1017/S1759078721000416

3. Об установлении зон ограничения полетов. Приказ Министерства транспорта РФ № 255.

4. Об установлении запретных зон. Приказ Министерства транспорта РФ № 254.

5. Полётные планы // URL: https://fpln.ru

6. Об утверждении федеральных правил использования воздушного пространства Российской Федерации. Постановление Правительства Российской Федерации № 138.

7. Showman G.A. An overview of radar imaging // Principles of Modern Radar. 2010. vol. I. Basic Principles. Pp. 835–891. SciTech Publishing. 2010.

8. **Hogan D., Brown J.** SAR 101: An introduction to synthetic aperture radar, Capella Space. 2020 // URL: https://www.capellaspace.com/sar-101-an-introduction-to-synthetic-aperture-radar

9. Adhikari G., Halder S., Banerjee S., Chaudhuri S.S. An effective use of synthetic aperture radar imaging technique over optical remote sensing and real aperture radar for mapping of Earth surface features // Innovations in Energy Management and Renewable Resources (52042). 2021. Pp. 1–6. DOI: 10.1109/IEM-RE52042.2021.9386850

10. **Slover Jason.** Synthetic aperture radar simulation by electro optical to SAR transformation using generative adversarial network. Thesis. Rochester Institute of Technology, 2020.

11. Sai Raghava Mukund Bhamidipati, Chinmaya Srivatsa, Chethan Kanakapura Shivabasave Gowda, Srikanth Vadada. Generation of SAR images using deep learning. SN Computer Science. 2020 // URL: https:// doi.org/10.1007/s42979-020-00364-z

12. **Balz T., Hammer H., Auer S.** Potentials and limitations of SAR image simulators – A comparative study of three simulation approaches // ISPRS Journal of Photogrammetry and Remote Sensing, 2015, vol. 101, Pp. 102–109. Available: https://doi.org/10.1016/j.isprsjprs.2014.12.008

13. **Balz T.** Low-cost real-time SAR simulation for applications in mission planning, education and information extraction // Proc. of the ISPRS Commission VII WG2 & WG7 Conf. on Information Extraction from SAR and Optical Data, with Emphasis on Developing Countries. Istanbul. 2007.

14. Shuttle Radar Topography Mission // URL: https://www.earthdata.nasa.gov/sensors/srtm

15. Auer S., Bamler R., Reinartz P. RaySAR – 3D SAR simulator: Now open source // 2016 IEEE Internat. Geoscience and Remote Sensing Symp. (IGARSS). 2016. Pp. 6730–6733. DOI: 10.1109/IG-ARSS.2016.7730757

16. Auer S.J. 3D synthetic aperture radar simulation for interpreting complex urban reflection scenarios. Technical University of Munich, 2011.

17. Hammer H., Schulz K. Dedicated SAR simulation tools for ATR and scene analysis // 2011, Proc. SPIE 8179, SAR Image Analysis, Modeling, and Techniques XI, 81790N. https://doi.org/10.1117/12.897452

18. Del Rosso M.P., Masini A., Bracci A., Ridolfi L., Cicciù F., Ullo S.L. Automatic processing chain for the generation of simplified sar images of large scenes // 2022 IEEE Internat. Geoscience and Remote Sensing Symp. 2022. Pp. 2955–2958. DOI: 10.1109/IGARSS46834.2022.9883859

19. Enomoto K., Sakurada K., Wang W., Kawaguchi N., Matsuoka M., Nakamura R. Image translation between SAR and optical imagery with generative adversarial nets // IEEE Internat. Geoscience and Remote Sensing Symp. 2018. Pp. 1752–1755. DOI: 10.1109/IGARSS.2018.8518719

20. Li Y., Fu R., Meng X., Jin W., Shao F. A SAR-to-Optical image translation method based on conditional generation adversarial network (cGAN) // IEEE Access. 2020. vol. 8. Pp. 60338–60343. 2020. DOI: 10.1109/ ACCESS.2020.2977103

21. Рекомендация MCЭ-R P.525-4, 2019 // URL: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.525-4-201908-I!!PDF-R.pdf.

22. Recommendation ITU-R P.2108-1, 2021 // URL: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.2108-1-202109-I!!PDF-E.pdf.

23. Рекомендация МСЭ-R BT. 601-7, 2011 // URL: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-R.pdf.

24. Intajag S., Chitwong S. Speckle noise estimation with generalized gamma distribution // SICE-ICASE Internat. Joint Conf. 2006. Pp. 1164–1167. DOI: 10.1109/SICE.2006.315296

25. Singh P., Diwakar M., Shankar A., et al. A review on SAR image and its despeckling. 2021, Arch Computat Methods Eng 28, 4633–4653 // Available: https://doi.org/10.1007/s11831-021-09548-z

26. **Vinayak R.K., Kong W., Valiant G., Kakade S.** Maximum likelihood estimation for learning populations of parameters // Proc. of the 36th Internat. Conf. on Machine Learning, 2019, PMLR 97: 6448–6457.

27. **Richards F.S.G.** A method of maximum-likelihood estimation // J. of the Royal Statistical Society: Series B (Methodological). 1961, 23: 469–475. https://doi.org/10.1111/j.2517-6161.1961.tb00430.x

28. Pan J.X., Fang K.T. Maximum likelihood estimation // Growth Curve Models and Statistical Diagnostics. Springer Series in Statistics. Springer, New York, NY, 2002. https://doi.org/10.1007/978-0-387-21812-0_3

29. **Rossi R.J.** Mathematical statistics: An introduction to likelihood based inference. New York: John Wiley & Sons, 2018. P. 227.

30. Wang B.H., Wang D.B., Ali Z.A., Ting B., Wang H. An overview of various kinds of wind effects on unmanned aerial vehicle // Measurement and Control. 2019. 52 (7-8): 731–739. DOI: 10.1177/0020294019847688

31. Wang B., Ali Z.A., Wang D. Controller for UAV to oppose different kinds of wind in the environment // J. of Control Science and Engineering. 2020. Article ID 5708970. 10 p. https://doi.org/10.1155/2020/5708970

32. Whitehead K., Hugenholtz C.H. Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: A review of progress and challenges // J. of Unmanned Vehicle Systems, 2014. https://doi. org/10.1139/juvs-2014-0006

REFERENCES

1. Keating E.G., Kerman J., Arthur D. Usage patterns and costs of unmanned aerial systems. Congressional budget office. 2021. Available: *http://www.cbo.gov/publication/57090*

2. Svedin J., Bernland A., Gustafsson A., Claar E., Luong J. Small UAV-based SAR system using low-cost radar, position, and attitude sensors with onboard imaging capability. *International Journal of Microwave and Wireless Technologies*, 2021, vol. 13 (6), Pp. 602–613. DOI: 10.1017/S1759078721000416

3. Ob ustanovlenii zon ogranicheniya poletov. Prikaz Ministerstva transporta RF No. 255. (rus)

4. Ob ustanovlenii zapretnykh zon. Prikaz Ministerstva transporta RF No. 254. (rus)

5. Poletnyye plany. Available: https://fpln.ru

6. Ob utverzhdenii federalnykh pravil ispolzovaniya vozdushnogo prostranstva rossiyskoy federatsii. Postanovleniye Pravitelstva Rossiyskoy Federatsii No. 138. (rus)

7. Showman G.A. An overview of radar imaging. *Principles of Modern Radar*, 2010, vol. I. Basic Principles, Pp. 835–891. SciTech Publishing, 2010.

8. Hogan D., Brown J. SAR 101: An introduction to synthetic aperture radar, Capella Space. 2020. Available: *https://www.capellaspace.com/sar-101-an-introduction-to-synthetic-aperture-radar*

9. Adhikari G., Halder S., Banerjee S., Chaudhuri S.S. An effective use of synthetic aperture radar imaging technique over optical remote sensing and real aperture radar for mapping of Earth surface features. Innovations in Energy Management and Renewable Resources (52042), 2021, Pp. 1–6. DOI: 10.1109/IEM-RE52042.2021.9386850

10. Slover Jason. Synthetic aperture radar simulation by electro optical to SAR transformation using generative adversarial network. Thesis. Rochester Institute of Technology. 2020.

11. Sai Raghava Mukund Bhamidipati, Chinmaya Srivatsa, Chethan Kanakapura Shivabasave Gowda, Srikanth Vadada. Generation of SAR images using deep learning. SN Computer Science. 2020. Available: https:// doi.org/10.1007/s42979-020-00364-z

12. **Balz T., Hammer H., Auer S.** Potentials and limitations of SAR image simulators – A comparative study of three simulation approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2015, vol. 101, Pp. 102–109. Available: https://doi.org/10.1016/j.isprsjprs.2014.12.008

Intellectual Systems and Technologies

13. **Balz T.** Low-cost real-time SAR simulation for applications in mission planning, education and information extraction. *Proceedings of the ISPRS Commission VII WG2 & WG7 Conference on Information Extraction from SAR and Optical Data, with Emphasis on Developing Countries,* Istanbul. 2007.

14. Shuttle Radar Topography Mission. Available: https://www.earthdata.nasa.gov/sensors/srtm

15. Auer S., Bamler R., Reinartz P. RaySAR – 3D SAR simulator: Now open source. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016, Pp. 6730–6733. DOI: 10.1109/IG-ARSS.2016.7730757

16. Auer S.J. 3D synthetic aperture radar simulation for interpreting complex urban reflection scenarios. Technical University of Munich, 2011.

17. Hammer H., Schulz K. Dedicated SAR simulation tools for ATR and scene analysis. 2011, Proc. SPIE 8179, SAR Image Analysis, Modeling, and Techniques XI, 81790N, https://doi.org/10.1117/12.897452

18. Del Rosso M.P., Masini A., Bracci A., Ridolfi L., Cicciù F., Ullo S.L. Automatic processing chain for the generation of simplified sar images of large scenes. *IEEE International Geoscience and Remote Sensing Symposium*, 2022, Pp. 2955–2958. DOI: 10.1109/IGARSS46834.2022.9883859

19. Enomoto K., Sakurada K., Wang W., Kawaguchi N., Matsuoka M., Nakamura R. Image translation between sar and optical imagery with generative adversarial nets. *IEEE International Geoscience and Remote Sensing Symposium*, 2018, Pp. 1752–1755. DOI: 10.1109/IGARSS.2018.8518719

20. Li Y., Fu R., Meng X., Jin W., Shao F. A SAR-to-Optical image translation method based on conditional generation adversarial network (cGAN). *IEEE Access*, 2020, vol. 8, Pp. 60338–60343. 2020. DOI: 10.1109/ACCESS.2020.2977103

21. Rekomendatsiya MSE-R P.525-4, 2019. Available: *https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.525-4-201908-1!!PDF-R.pdf*. (rus)

22. Recommendation ITU-R P.2108-1, 2021. Available: *https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.2108-1-202109-I!!PDF-E.pdf*

23. Rekomendatsiya MSE-R BT. 601-7, 2011. Available: *https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-R.pdf*. (rus)

24. Intajag S., Chitwong S. Speckle noise estimation with generalized gamma distribution. *SICE-ICASE International Joint Conference*, 2006, Pp. 1164–1167. DOI: 10.1109/SICE.2006.315296

25. Singh P., Diwakar M., Shankar A., et al. A review on SAR image and its despeckling. 2021, Arch Computat Methods Eng 28, 4633–4653. Available: *https://doi.org/10.1007/s11831-021-09548-z*

26. Vinayak R.K., Kong W., Valiant G., Kakade S. Maximum likelihood estimation for learning populations of parameters. *Proceedings of the 36th International Conference on Machine Learning*, 2019, PMLR 97:6448-6457.

27. **Richards F.S.G.** A method of maximum-likelihood estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1961, 23: 469–475. https://doi.org/10.1111/j.2517-6161.1961.tb00430.x

28. Pan J.X., Fang K.T. Maximum likelihood estimation. *Growth Curve Models and Statistical Diagnostics*. Springer Series in Statistics. Springer, New York, NY, 2002. https://doi.org/10.1007/978-0-387-21812-0_3

29. Rossi R.J. *Mathematical statistics: An introduction to likelihood based inference*. New York: John Wiley & Sons, 2018. P. 227.

30. Wang B.H., Wang D.B., Ali Z.A., Ting B., Wang H. An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control*, 2019, 52 (7-8): 731–739. DOI: 10.1177/0020294019847688

31. Wang B., Ali Z.A., Wang D. Controller for UAV to oppose different kinds of wind in the environment. *Journal of Control Science and Engineering*, 2020, Article ID 5708970, 10 p. https://doi.org/10.1155/2020/5708970

32. Whitehead K., Hugenholtz C.H. Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: A review of progress and challenges. *Journal of Unmanned Vehicle Systems*, 2014. https://doi.org/10.1139/juvs-2014-0006

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Прокофьев Кирилл Владимирович Kirill V. Prokofiev E-mail: baterflyrity@yandex.ru

Иванов Сергей Владимирович Sergey V. Ivanov E-mail: sergei.v.ivanov@gmail.com

Поступила: 27.09.2022; Одобрена: 16.12.2022; Принята: 12.01.2023. Submitted: 27.09.2022; Approved: 16.12.2022; Accepted: 12.01.2023.

CC OS

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15402 УДК 681.5

ФОРМИРОВАНИЕ УПРАВЛЕНИЯ ПОЛЕТОМ ГРУППЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ НА ОСНОВЕ АЛГОРИТМА МНОГОАГЕНТНОЙ МОДЕЛИ РОЕНИЯ

Ю. Чжу≀ ⊠

¹ Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация

^{III} 1918149382@qq.com

Аннотация. Рассмотрена задача управления группой беспилотных летательных аппаратов (БПЛА) для организации движения роем по заданной траектории, обеспечивающей наиболее эффективное достижение цели полета. Изучены вопросы выбора математической модели пространственного движения группы БПЛА, пригодной для решения задачи синтеза согласованного управления всей совокупностью летательных аппаратов. Учитывая специфику требований к пространственно-временному положению отдельных БПЛА (агентов) в группе, предложено использовать модель с лидером. В группе агентов находится виртуальный лидер, который планирует маршрут движения группы в соответствии с заданной задачей и отслеживает определенную цель движения. Виртуальный лидер рассчитывает свое собственное управление движением с помощью алгоритма отслеживания траектории или отслеживания цели, чтобы двигаться по желаемой траектории. При этом сигнал наведения может позволить отдельным БПЛА собираться на позиции виртуального лидера и соответствовать вектору скорости виртуального лидера, чтобы обеспечить связь топологии многоагентной системы и обеспечить роевое формирование.

Ключевые слова: система управления, беспилотный летательный аппарат, математическая модель группы, координация, управление роем

Для цитирования: Чжу Ю. Формирование управления полетом группы беспилотных летательных аппаратов на основе алгоритма многоагентной модели роения // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 22–36. DOI: 10.18721/JCSTCS.15402 Research article DOI: https://doi.org/10.18721/JCSTCS.15402 UDC 681.5



FORMATION OF FLIGHT CONTROL FOR A GROUP OF UNMANNED AERIAL VEHICLES BASED ON ALGORITHM OF MULTI-AGENT SWARM MODEL

Y. Zhu¹ ⊠

¹ Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation □ 1918149382@qq.com

Abstract. The problem of controlling a group of unmanned aerial vehicles (UAVs) is considered to organize the movement of a swarm along a given trajectory, which ensures the most effective achievement of the flight goal. The issues of choosing a mathematical model of the spatial motion of a group of UAVs, suitable for solving the problem of synthesis of coordinated control of the entire set of aircrafts, are discussed. Taking into account the specifics of the requirements for the space-time position of individual UAVs (agents) in a group, it is proposed to use a model with a leader. A group of agents has a virtual leader who plans the route of the group in accordance with a given task and tracks a specific goal of movement. The virtual leader calculates its own motion control with a trajectory-tracking or target-tracking algorithm to move along the desired trajectory. In this case, the guidance signal can allow individual UAVs to gather at the position of the virtual leader and correspond to the velocity vector of the virtual leader in order to communicate the topology of the multi-agent system and ensure swarm formation.

Keywords: control system, unmanned aerial vehicle, mathematical model of the group, coordination, swarm management

Citation: Zhu Y. Formation of flight control for a group of unmanned aerial vehicles based on algorithm of multi-agent swarm model. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 22–36. DOI: 10.18721/JCSTCS.15402

Введение

Успехи, достигнутые в разработке беспилотных летательных аппаратов (БПЛА) как военного, так и гражданского назначения, создают хорошие предпосылки для широкого круга их применения, в частности для исследования земной поверхности, в том числе зон стихийных бедствий, мест чрезвычайных ситуаций, для обнаружения очагов пожаров, для измерения метеорологических данных, радиационного фона и других факторов, а также для обеспечения телекоммуникаций, мониторинга трубопроводов и линий электропередач, патрулирования границ. Связано это с тем, что БПЛА дешевле пилотируемой авиации, более простые в обслуживании, их полет может быть намного продолжительнее, они также могут работать в темноте, в условиях плохой видимости, кроме того, они могут применяться в ситуациях, угрожающих жизни пилота.

Эффективность использования БПЛА значительно повышается при организации групповых полетов. Однако при этом возникает ряд сложных проблем, связанных с обеспечением управления полетом группы БПЛА. Под группой БПЛА обычно понимают некоторую совокупность летательных аппаратов, подчиняющихся определенным правилам сбора в группу, способных выдерживать свое место в строю на прямолинейных и криволинейных участках полета всей группы в целом, реагировать на изменения окружающей среды и взаимодействовать друг с другом для решения единой целевой задачи, поставленной перед группой.

Это определяет необходимость гибкого формирования группы в операции: позволяет выбирать параметры пространственно-временного и комбинаторного взаимного положения элементов группы, исходя из требований и целей операции.

С точки зрения пространственно-временного положения элементов в группе, последние могут выполнять:

• групповой маневр, при котором номинальные траектории движения всех элементов конгруэнтны;

• индивидуальный маневр, при котором каждый элемент, независимо от других, может менять параметры траектории движения;

• смешанный маневр, при котором возможно изменение параметров движения как всей группой средств, так и отдельными элементами.

Что касается комбинаторного положения, то оно характеризует структуру построения группы БПЛА различных типов, например, целевых элементов (непосредственно решающих задачу) и обеспечивающих её выполнение БПЛА. Формирование пространственно-временной модели построения группы должно базироваться на учете ограничений по располагаемой энергетике, маневренным возможностям, взаимной расстановке и т. д.

При этом на систему управления полетом группы БПЛА возлагается задача согласованного управления каждым из летательных аппаратов для организации движения строем по заданной траектории, которая обеспечивает наиболее эффективное достижение цели, поставленной перед группой. Система управления полетом БПЛА в группе осуществляет контроль правильности удержания траектории каждым БПЛА с точностью, обеспечивающей безопасность в плотных групповых порядках. На эту же систему возлагается задача межсамолетной навигации и организации взаимодействия БПЛА по информационным каналам с целью определения взаимных координат [1].

В области исследований формирование группы БПЛА постепенно становится новой разнонаправленной и междисциплинарной областью исследований. В этой области существует множество направлений исследований: исследования аэродинамических помех при близком формировании группы [2–4], исследования по проектированию формирования группы [5–7], исследования по планированию траектории формирования группы [8, 9], исследования по управлению формированием группы [10–13], исследования по реконструкции формирования группы [14–16], исследования по распределению задач формирования группы [17, 18], предотвращение столкновений БПЛА и т. д. Поскольку управление формированием строя (группы, роя) является важной технологией, то в этой области многочисленные исследователи предложили большое количество алгоритмов управления формированием группы. При этом теоретическая основа алгоритмов группового управления как правило связана с моделью роевого движения.

Для достижения высокой стабильности и хорошей координации группы БПЛА получается, что управление формированием групп БПЛА сложнее управления одним БПЛА (агентом). Необходимо учитывать не только контроль над мультиагентом (членом группы), но и принимать интерактивные и совместные решения по получаемой информации для обеспечения эффективного управления. Необходимо также учитывать взаимодействие между агентами и взаимодействие между агентами и окружающей средой.

Полет группы БПЛА можно представить как движение роя. Модель роевого движения — модель, используемая для изучения движения одновременно большого количества индивидуумов. Формулируя закон индивидуального движения, можно определить относительное регулярное положение между перемещениями отдельных БПЛА, при этом движения индивидуального БПЛА имеют тенденцию приближаться к регулярности.

Следовательно, необходимо не только учитывать совместное управление мультиагентами, но и изучать модель роя и движение строя мультиагентов.

В статье предлагается модель роя в применении к многоагентной системе с кооперативным движением: модель движение роя.

Модель роя БПЛА

Рой — повсеместное явление в природе. Это врожденная способность, приобретенная после длительной эволюции социальной группы организмов, объединяться в группы и тем самым адаптироваться к окружающей среде для выживания при восполнении слабости отдельных членов группы. Примерами могут служить стаи птиц, рыб и др.

В 1986 году Рейнольдс предложил три эвристических правила, которые привели к созданию первой компьютерной анимации группового поведения [19]:

1) Центрирование стаи: старайтесь держаться ближе к ближайшим товарищам по рою.

2) Предотвращение столкновений: избегайте столкновений с ближайшими товарищами по рою.

3) Сопоставление скорости: попытка сопоставить скорость с ближайшими товарищами по рою.

Эти правила также известны в литературе как правила сплоченности, разделения и согласования. Исходя из этих правил различными исследователями было предложено множество моделей роя.

В 1995 году Vicsek предложил модель роя многоагентной сложной системы и создал простую модель группового движения — модель Vicsek на основе статистической механики [20, 26]. Джадбабайе и др. [22] представили последовательность графов соседей, образованную всеми индивидуальными отношениями местоположения в модели Vicsek, и теоретически доказали, что, если эти графы соседей связаны согласованным образом, группа будет синхронизирована. Эта модель описывает синхронное движение автономной системы, состоящей из нескольких агентов. Агенты в модели следуют следующим правилам:

1. Агент, движущийся в системе, имеет постоянную скорость *p*.

2. Любая пара агентов в системе имеет радиус влияния *r*. Только когда прямолинейное расстояние между парой агентов меньше *r*, они могут влиять друг на друга.

3. Направление движения агента в каждый момент такое же, как и среднее направление движения всех других агентов в пределах радиуса влияния в предыдущий момент.

В этой модели агент *i* имеет постоянную скорость *p*, смещение равно $q_i(t)$, а направление скорости агента *i* равно $\theta_i(t)$, что удовлетворяет следующим условиям:

$$\theta_i(t+1) = \frac{1}{n_i(t)} \sum_{j \in N_i(t)} \theta_j(t), \qquad (1)$$

где $N_i(t) = \{j < k | \| q_i(t) - q_j(t) \| \le r\}.$

Кукер и Смейл предложили кластерную модель группы (CS-модель) [23], основанную на работе Vicsek. Модель Couzin [27] представила проблему перспективного угла, основанную на модели Vicsek. Первоначальное намерение авторов [27] состояло в том, чтобы моделировать группу животных с учетом ограниченного пространства и улучшить эффективность синхронизации модели группы. Этот результат исследования показывает, что агент в базовой модели Vicsek использует информацию о соседних агентах избыточно, т. е. агенту не нужно получать всю информацию о соседях в группе и обладать информацией о всех членах группы. Увеличение числа членов группы не обязательно означает повышение эффективности синхронизации.

Данная модель имеет следующие допущения:

1. Все агенты имеют взаимное влияние.

2. Значимость агента, на которого действуют другие агенты, зависит от абсолютного расстояния и разницы в скорости между ними.

Непрерывная модель CS имеет многоагентную систему с N агентами, где смещение агента i в момент времени t записывается как $q_i(t)$, а скорость записывается как $p_i(t)$:

$$\dot{q}_i(t) = p_i(t) \tag{2}$$

$$\dot{p}_{i}(t) = \alpha \sum_{j \in N, j \neq i} a_{ij} \left(\left\| q_{j}(t) - q_{i}(t) \right\| \right) \left(p_{j}(t) - p_{i}(t) \right),$$
(3)

где $a_{ij}(t) = \frac{1}{N} \phi(||q_j(t) - q_i(t)||),$ и $\phi(r) = \frac{1}{(1 + ||r||^2)^{\beta}},$ параметры $\alpha > 0, \beta > 0.$

В исследовании Баллерини и др. [24] было обнаружено, что особи в стае птиц взаимодействуют только с ближайшими 6-7 соседями во время движения стати и это не связано с плотностью группы и расстоянием до птиц-соседей. Кроме того, было принято, что в зависимости от способности отдельного организма запоминать информацию, возможно, что запоминаемая информация оказывает немаловажное влияние на принятие решений о временном поведении агента.

Джордж [25] считает, что, если запоминаемая информация о соседях может быть эффективно использована, то скорость сходимости синхронизации группы также будет улучшена.

Следовательно, можно считать возможным построить частично наблюдаемую групповую модель (модель роя) для управления групповым движением (движением роя).

Тадмор и др. создали новую модель кластера многоагентных сложных систем (МТ), основанную на модели CS [28]. Непрерывная модель МТ рассматривает сложную систему с N индивидуумами. Для индивидуума *i* в момент времени *t* смещение записывается как $q_i(t)$, а скорость записывается как $p_i(t)$, то $\lceil q_i(t), p_i(t) \rceil$ удовлетворяет:

$$\dot{q}_i(t) = p_i(t) \tag{4}$$

$$\dot{p}_{i}(t) = \alpha \sum_{j \in N, j \neq i} b_{ij} \left(\left\| q_{j}(t) - q_{i}(t) \right\| \right) \left(p_{j}(t) - p_{i}(t) \right),$$
(5)

где
$$b_{ij}(t) = \frac{\varphi(\|q_i(t) - q_i(t)\|)}{\sum_{k \in N} \varphi(\|q_k(t) - q_i(t)\|)}$$
, и $\varphi(r) = \frac{1}{(1 + \|r\|^2)^{\beta}}$, параметры $\alpha > 0, \beta > 0$.

На основе модели CS Шен и др. [29] предложили модель роя с механизмом лидерства (модель HL). Модель HL рассматривает сложную иерархическую систему, состоящую из (k + 1) агентов и индивидов [0, 1, 2, ..., k]. При этом каждый уровень соответствует соответствующему индивиду, $a_{ij}(t)$ означает, что индивид *i* возглавляет индивид *j*, что удовлетворяет двум условиям:

1.
$$j \le i$$
, to $a_{ii}(t) \ne 0$;

2. если набор лидеров индивидуума *i* равен $L = \{a_{ij}(t) > 0\}$, то для любого i > 0 есть $L \neq \Phi$.

Ли Чжучунь [30] дополнительно улучшил модель HL на основе модели Шэнь и установил более общую модель кластера многоагентной сложной системы с единым механизмом лидерства, которая имеет глобального лидера и не зависит от других агентов.

Алгоритм управления движением роя по модели Олфати-Сэйбера. Олфати-Сэйбер [29] разработал алгоритм роения с несколькими БПЛА, имея ввиду, что роение — скопление БПЛА, которое относится к большому количеству БПЛА, движущихся вместе. В природе коллективное движение стаей пчел, птиц, рыб и животных является типичным примером роения. Формирование роящейся группы часто не имеет глобального лидера или информационного центра для обработки и выдачи групповой информации. Агенты в группе могут координировать общее поведение группы в соответствии с простыми местными правилами. Алгоритм разделяет силу агента на условия искусственного потенциального поля, условия согласования скорости и условия наведения. Алгоритм мультиагентного роения используется для вычисления вектора позиции, и получается распределенная, постепенно генерируемая формация квазирешетки, которая имеет хорошую надежность и масштабируемость, чтобы поддерживать группу роя агентов и формы, тем самым обеспечивая предотвращение столкновения с препятствиями и отслеживание.

Методы моделирования и анализа роения включают в себя метод Лагранжа, метод Эйлера и метод дискретной модели системы. Метод Лагранжа использует обыкновенные дифференциальные уравнения для описания всех индивидов в группе, которые могут описывать процесс динамического изменения объекта; метод Эйлера описывает группу как поле в пространстве, а функция плотности поля представляет индивидов в группе. Плотность распределения в пространстве выражается уравнениями в частных производных, чтобы выразить явление диффузии поля плотности, тем самым отражая динамику групповых изменений. Модель дискретной системы принимает вид набора уравнений дискретного времени для выражения динамики системы.

Алгоритм, используемый в модели Олфати-Сэйбера, выглядит следующим образом.

В трехмерном евклидовом пространстве динамика агента *i* моделируется как интегральное звено второго порядка:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases}, \ i = 1, \ 2, \ \dots, \ N,$$
(6)

где q_i, p^i и $u_i \in R^n$ представляют положение, скорость и управляющий вход *i*-го агента соответственно. Каждый агент может взаимодействовать только с соседними агентами в своей области связи, и в момент времени *t* набор соседних агентов может быть выражен следующим образом:

$$N_{i}^{\alpha}(t) = \left\{ j : \left\| q_{i} - q_{j} \right\| \le r, \ j = 1, \ 2, \ \dots N, \ j \neq i \right\},$$
(7)

где $\|q_i - q_j\|$ – расстояние Эйлера, а r – максимальный радиус взаимодействия или максимальное критическое расстояние. Желаемая геометрическая модель кластера требует, чтобы каждый агент находился на одинаковом расстоянии от всех соседних индивидуумов и удовлетворял следующим ограничениям:

$$\left\|\boldsymbol{q}_{i}-\boldsymbol{q}_{j}\right\|=d,\;\forall i,\;j\in N_{i}\left(t\right),\tag{8}$$

где d — положительная константа, указывающая минимально допустимое расстояние или минимальное критическое расстояние между каждой парой соседних агентов, и $d \le r$.

В среде с множеством препятствий входные данные каждого агента в многоагентном алгоритме управления состоят из следующих трех частей [29]:

$$u_i = u_i^{\alpha} + u_i^{\beta} + u_i^{\gamma}, \qquad (9)$$

где α, β, у представляют трех агентов, основанных на теории роения Олфати-Сэйбера.

Агент α представляет любого агента в группе, агент β генерируется проекцией соседнего агента α на поверхность препятствия, используемую для представления физического препятствия, которое необходимо обойти, а агент γ используется для построения навигационной обратной связи,

указывающей цель, которую необходимо отслеживать. u_i^{α} означает элементы взаимодействия (α , α), u_i^{β} – элементы взаимодействия (α , β), а u_i^{γ} – распределенная навигационная обратная связь. Определения u_i^{α} , u_i^{β} и u_i^{γ} следующие:

$$u_i^{\alpha} = -c_q^{\alpha} \sum_{j \in N_i^{\alpha}} \rho_H(q_i) \varphi_{\alpha}(q_i) - c_v^{\alpha} \sum_{j \in N_i^{\alpha}} a_{ij}(q_i) (p_i - p_j), \qquad (10)$$

$$u_{i}^{\beta} = -c_{q}^{\beta} \sum_{k \in N_{i}^{\beta}} b_{i,k}\left(q_{i}\right) \varphi_{\beta}\left(q_{i}\right) - c_{\nu}^{\beta} \sum_{k \in N_{i}^{\beta}} b_{i,k}\left(q_{i}\right) \left(p_{i} - \hat{p}_{i,k}\right), \tag{11}$$

$$u_i^{\gamma} = -c_q^{\gamma} \varphi_{\gamma} \left(q_i - q_{\gamma} \right) - c_p^{\gamma} \left(p_i - p_{\gamma} \right), \tag{12}$$

 u_i^{α} состоит из двух частей. Первая часть устанавливает расстояние между агентами как желаемое расстояние, а вторая часть делает скорость агентов согласованной со скоростью соседних агентов. Конкретное выражение первой части заключается в следующем:

$$z_{ij} = (q_i - q_j) - \frac{q_i - q_j}{\|q_i - q_j\|} * d,$$
(13)

$$\varphi_{\alpha}\left(q_{i}\right) = \frac{z_{ij}}{\sqrt{1 + \epsilon_{\alpha} \left\|z_{ij}\right\|^{2}}},\tag{14}$$

$$\rho_{H}(q_{i}) = \frac{\left(\left\|q_{i} - q_{j}\right\| - d\right)^{2}}{H} + 1,$$
(15)

где H, ϵ_{α} , c_{q}^{α} и c_{p}^{α} являются нормальными числами, а значение H обычно больше, чем d. Фрагментация – это алгоритм кластеризации Олфати-Сэйбера, и введение $\rho_{H}(q_{i})$ может эффективно предотвратить фрагментацию. Когда расстояние между агентами увеличивается, значение $\rho_{H}(q_{i})$ также быстро увеличивается.

Второй компонент
$$u_i^{\alpha}$$
 равен $a_{ij}(q_i) = \rho_h\left(\frac{\|q_i - q_j\|}{r}, h_{\alpha}\right) \in [0, 1], j \neq i. a_{ij} \equiv 0$, если $a_{ij} \neq 0$,

обмен информацией между агентами *i* и *j* также может рассматриваться как обмен информацией между агентами.

 $\rho_h(z)$ – функция воздействия, как показано ниже:

$$\rho_{h}(z) = \begin{cases} 1, & z \in [0, h] \\ \frac{1}{2} \left[1 + \cos\left(\pi \frac{(z-h)}{(1-h)}\right) \right], & z \in [h, 1] \\ 0, & otherwise \end{cases}$$
(16)

 u_i^{γ} позволяет агенту отслеживать виртуального пилота или желаемую траекторию, c_q^{γ} , c_p^{γ} – обычное число, q_{γ} и p_{γ} представляют положение и скорость виртуального пилота соответственно, $\phi_{\gamma}(q_i - q_{\gamma})$ имеет следующее представление:

$$\varphi_{\gamma}\left(q_{i}-q_{\gamma}\right) = \frac{q_{i}-q_{\gamma}}{\sqrt{1+\epsilon_{\gamma}\left\|q_{i}-q_{\gamma}\right\|^{2}}}$$
(17)

 u_i^{β} позволяет агенту обходить препятствия, где c_q^{β} и c_p^{β} являются нормальными числами. Принцип заключается в следующем: виртуальный β - агент с положением и скоростью создается на поверхности препятствия в пределах диапазона обнаружения агента в кластере. Метод построения заключается в следующем [29].

1. Для совпадения с границей гиперплоскости и единичной нормалью a_k , проходящей через точку y_k , положение и скорость β - агента определяются по следующей формуле:

$$\hat{q}_{i,k} = Pq_i + (I - P)y_k, \ \hat{p}_{i,k} = Pp_i,$$
(18)

где $P = I - a_k a_k^T$ – проекционная матрица.

2. Для сферического препятствия с радиусом R_k и центральной точкой y_k , положение и скорость β - агента равны:

$$\hat{q}_{i,k} = \mu q_i + (I - \mu) y_k, \ \hat{p}_{i,k} = \mu P p_i,$$
(19)

где $\mu = \frac{R_k}{\|q_i - y_k\|}, \ a_k = \frac{q_i - y_k}{\|q_i - y_k\|}, \ P = I - a_k a_k^T.$

Созданный таким образом виртуальный β- агент, направлен на то, чтобы индивидуальная скорость в группе соответствовала скорости виртуального β- агента, сохраняя при этом определенную дистанцию.

 $\phi_{\scriptscriptstyle B}(q_i)$ и $b_{i,k}(q_i)$ в $u_i^{\scriptscriptstyle B}$ определяются как

$$\varphi_{\beta}(q_{i}) = \frac{q_{i} - \hat{q}_{i,k}}{\sqrt{1 + \epsilon_{\beta} \left\| q_{i} - \hat{q}_{i,k} \right\|^{2}}} - 1,$$
(20)

$$b_{i,k}\left(q_{i}\right) = \rho_{h}\left(\frac{\left\|q_{i}-\hat{q}_{i,k}\right\|}{r_{o}},h_{\beta}\right),\tag{21}$$

где ϵ_{β} – нормальное число, а r_o – максимальное расстояние обнаружения БПЛА относительно препятствия.

Упрощенный алгоритм многоагентной модели роения

Конструкция Олфати-Сэйбера имеет множество параметров искусственного потенциального поля и характеризуется громоздкими вычислениями. Принимая во внимание требования системы управления БПЛА к вычислительной эффективности, было проведено упрощение модели Олфати-Сэйбера с учетом только условия искусственного потенциального поля и условия согласования скоростей и условия наведения.

Решетчатые и квазирешеточные структуры. В многоагентной системе если расстояние между каждым агентом и соседним агентом одинаково, относительное положение многоагентной системы составляет решетчатую структуру.

Если расстояние между агентом и соседним агентом не может быть точно одинаковым, а изменяется в пределах определенного диапазона, то относительные позиционные отношения мультиагента образуют структуру квазирешетки. Для квазирешеточной структуры расстояние между соседними агентами примерно одинаково.

Разработка многоагентного алгоритма роения. Алгоритм многоагентного роения предполагает такой управляющий ввод u_i , который заставит относительные пространственные положения мультиагентов образовывать структуру, подобную решетке или квазирешетке, и сводить вектор скорости мультиагентов к согласованному значению. Алгоритм Олфати-Сэйбера представляет элемент управления u_i *i*-го агента как комбинированное действие трех элементов управления:

$$u_i = u_i^g + u_i^d + u_i^\gamma, \tag{22}$$

где u_i^g – элемент искусственного потенциального поля, используемый для реализации агрегации и предотвращения столкновений агентов; u_i^d – элемент сопоставления скорости, используемый для реализации сопоставления скорости; u_i^γ – элемент руководства, используемый для отслеживания виртуального лидера, чтобы группа агентов двигалась в соответствии с желаемым направлением.

Эффекты искусственного потенциального поля и сопоставления скоростей отражают три принципа роения Рейнольдса. Однако если начальное положение группы агентов распределено случайным образом, топология связи системы может быть несвязанной, что приведет к разделению группы. Эффект руководства позволяет всем агентам отслеживать одного и того же виртуального лидера, а группа агентов собирается от начальной позиции до виртуального лидера, увеличивая связность топологии системы и избегая разделения групп.

Проектирование искусственного потенциального поля. Существует множество параметров искусственного потенциального поля, первоначально разработанных в модели Олфати-Сэйбера, что привело к громоздкому расчету. Для преодоления этого недостатка искусственное потенциальное поле было переработано, физическое значение при этом приобрело более четкий характер, параметры стали лучше отлажены, а расчет силы потенциального поля стал более лаконичен.

При этом функция инструмента проектирования заключается в следующем.

Функция плавного шага второго порядка $s_1(r)$ и $s_2(r)$

$$s_{1}(r) = \begin{cases} 1, \ r < r_{a} \\ \rho_{h}(z), \ z = \frac{r - r_{a}}{r_{b} - r_{a}}, \ r_{a} \le r \le r_{b}, \\ 0, \ r > r_{b} \end{cases}$$
(23)

$$s_2(r) = 1 - s_1(r),$$
 (24)

где r_a , r_b – левая и правая границы интервала шагов; $\rho_h(z)$ должна быть гладкой функцией второго порядка, зависит от детальных требований и граничных условий, например:

$$\rho_h(z) = \frac{1}{2} + \frac{3^n}{2(3^n - 1)} \cos \pi z - \frac{1}{2(3^n - 1)} \cos 3\pi z.$$
(25)

Представим функцию плавного шага второго порядка $S_3(r)$ в виде

$$s_{3}(r) = s_{1}(r)r + (r_{a} + r_{b})s_{2}(r)/2, \qquad (26)$$

где r_a, r_b используется для настройки интервального диапазона функции насыщения.

Функция силы искусственного потенциального поля, предназначенная для имитации линейной структуры, выглядит следующим образом:

$$f(r) = \frac{f_0}{r_0} (r - r_0) s_1(r), \qquad (27)$$

где r — относительное расстояние между двумя соседними агентами; r_0 — расстояние действия, когда сила искусственного потенциального поля равна нулю; f_0 — амплитуда силы потенциального поля, когда относительное расстояние между двумя соседними агентами равно нулю. Степенная функция может быть получена путем интегрирования силовой функции следующим образом:

$$\varphi(r) = \int_{r_0}^r f(z) \, dz. \tag{28}$$

При $r = r_0$ сила потенциального поля $f(r_0) = 0$, соседний агент находится в равновесии, а энергия потенциального поля $\phi(r_0) = 0$ имеет минимальное значение. При $r < r_0$, $f(r) \approx \epsilon f_0/r_0 \cdot (r - r_0) < 0$, сила потенциального поля пропорциональна относительному расстоянию между агентами и проявляется как сила отталкивания. Если $\phi(r) \approx f_0/2r_0 \cdot (r - r_0)^2$, то относительная сила между агентами аналогична упругости отталкивания. При $r > r_0$, в меньшем диапазоне, f(r) > 0, это проявляется как относительное притяжение между агентами, которое используется для организации образований роя; в большем диапазоне f(r) сходится к нулю, чтобы избежать взаимного вмешательства между агентами, не являющимися соседями, и потенциальная функция сходится к значению e_0 , которое представляет минимальную энергию для разделения соседних агентов.

Таким образом, функция $f(r_{\sigma})$ силы искусственного потенциального поля и потенциальная функция $\phi(r_{\sigma})$ выглядят следующим образом:

$$\begin{cases} f(r) = \frac{f_0}{r_0} (r - r_0) s_1(r), \\ \phi(r) = \int_{r_0}^r f(z) dz. \end{cases}$$
(29)

Элементы искусственного потенциального поля. Пусть *i*-й агент находится в искусственном потенциальном поле, генерируемом *j*-м агентом, и потенциальная функция выражается следующим образом:

$$\varphi_{i,j}\left(\left\|\boldsymbol{q}_{j}-\boldsymbol{q}_{i}\right\|\right)=\varphi\left(\left\|\boldsymbol{q}_{j}-\boldsymbol{q}_{i}\right\|\right),\tag{30}$$

где $\| * \|$ представляет значение векторного модуля. Потенциальная сила поля, создаваемая *i*-м агентом от *j*-м агентом, представляет собой отрицательный градиент энергии потенциального поля в положении *i*-го агента:

$$f_{i,j}(||q_{j}-q_{i}||) = -\nabla_{q_{i}}\varphi_{i,j}(||q_{j}-q_{i}||) = f(||q_{j}-q_{i}||)n_{i,j},$$
(31)

где $n_{i,j}$ – единичный вектор, представляющий положительное направление потенциальной силы поля, от *i*-го агента к *j*-му агенту; $f(||q_j - q_i||)$ – амплитуда потенциальной силы поля. Если расстояние между *i*-м агентом и *j*-м агентом мало, амплитуда отрицательна, и потенциальная

сила поля *i*-го агента направлена от *j*-го агента к *i*-му агенту, что является силой отталкивания. Если расстояние между *i*-м агентом и *j*-м агентом велико, амплитуда положительна, и потенциальная сила поля *i*-го агента направлена от *i*-го агента к *j*-му агенту, что является силой гравитации.

Если *i*-й агент находится в группе агентов, полученная потенциальная сила поля представляет собой совокупную силу потенциальных сил поля, создаваемых всеми другими соседними агентами, которая выражается следующим образом:

$$f_{i}^{g} = \sum_{j \in N_{i}} f_{i,j} \left(\left\| q_{j} - q_{i} \right\| \right).$$
(32)

Согласование скорости агентов. Роль согласования скорости заключается в том, чтобы группа агентов имела одинаковую скорость движения. Агент динамически регулирует свою собственную скорость на основе информации о скорости соседнего агента, вычисляет суммарную скорость соседнего агента на основе принципа векторной суперпозиции и использует её среднее значение в качестве собственной желаемой скорости. В фактическом расчете агент вычисляет разницу между соседним агентом и его собственным вектором скорости и принимает векторную сумму разницы в качестве члена соответствия скорости, который выражается следующим образом:

$$f_i^d = \sum_{j \in N_i} a_{ij}^d \left(q_i, \ q_j \right) \left(p_j - p_i \right), \tag{33}$$

где $a_{ij}^d(q_i, q_j)$ – весовой коэффициент, представляющий собой плавную ступенчатую функцию второго порядка относительного расстояния агента

$$a_{ij}^{d}(q_{i}, q_{j}) = s_{1}(||q_{j} - q_{i}||).$$
(34)

При перемещении группы агентов изменяется относительное положение группы, и соответственно изменяется отношение соседей между агентами. Когда один агент входит в соседство с другим агентом и выходит из него, эффект члена согласования скорости возникает или исчезает. Благодаря функции плавного шага второго порядка этот процесс изменения может быть сглажен. Можно избежать силовых мутаций и формировать способность поддержания стабильности.

Руководство движением роя. Предположим, что в группе агентов есть виртуальный лидер, который планирует маршрут движения группы в соответствии с заданной задачей или отслеживает определенную цель движения. Виртуальный лидер рассчитывает свое собственное управление с помощью алгоритма отслеживания траектории или отслеживания цели, чтобы двигаться по желаемой траектории. Модель виртуального лидера выглядит следующим образом:

$$\begin{cases} \dot{q}^{leader} = p^{leader} \\ \dot{p}^{leader} = u^{leader} \end{cases}, \tag{35}$$

где $q^{leader} \in R^3$ – вектор положения виртуального лидера; $p^{leader} \in R^3$ – вектор скорости виртуального лидера; u^{leader} – вектор управления виртуального лидера. Виртуальный лидер передает информацию о своем местоположении и скорости группе агентов во время движения, и группа агентов подчиняется виртуальному лидеру с помощью элемента управления, чтобы сформировать рой вокруг виртуального лидера.

Тогда руководящий сигнал алгоритма роения выражается как

$$f_i^{leader} = c_1^{leader} \left(q^{leader}, q_i \right) + c_2^{leader} \left(p^{leader} - p_i \right), \tag{36}$$

где $c_1^{leader}(q^{leader}, q_i)$ — часть согласования положения, которая является функцией плавного насыщения второго порядка; $c_2^{leader}(p^{leader} - p_i)$ — часть согласования скорости и c_2^{leader} — постоянный коэффициент; $c_1^{leader}(q^{leader}, q_i)$ определяется следующим образом:

$$c_{1}^{leader}\left(q^{leader}, q_{i}\right) = c_{1}^{leader}s_{3}\left(\left\|q^{leader} - q_{i}\right\|\right)\frac{q^{leader} - q_{i}}{\left\|q^{leader} - q_{i}\right\|},\tag{37}$$

где c_1^{leader} – коэффициент, соответствующий положению; $S_3(*)$ – функция насыщения второго порядка; $\|q^{leader} - q_i\|$ – расстояние между агентом и виртуальным лидером.

Сигнал наведения может позволить агентам собираться на позиции виртуального лидера и соответствовать вектору скорости виртуального лидера, чтобы обеспечить связь топологии связи многоагентной системы и сформировать роевое формирование. С другой стороны, из-за притяжения направляющего агента (лидера), который влияет на силу агента, исходная решетчатая структура образования между мультиагентами может нарушать квазирешетчатую структуру образования. Причина в том, что сила, создаваемая согласованием положения в термине наведения, пропорциональна расстоянию между агентом и виртуальным лидером. Агенты на периферии группы агентов подвергаются большему усилию и сжимаются внутри формирования, разрушая однородность формирования. Выражение части определения положения, соответствующей положению, в качестве функции насыщения может помочь избежать чрезмерного воздействия на периферийные агенты и вызвать помехи в формировании.

Основываясь на всеобъемлющей модели агента, терминах искусственного потенциального поля, терминах согласования скорости и терминах руководства, представление роевого формирования многоагентных систем выглядит следующим образом:

$$\begin{cases} \dot{q}^{leader} = p^{leader} \\ \dot{p}^{leader} = -\nabla_{q_i} \sum_{j \in N_i} \varphi_{i,j} \left(\left\| q_j - q_i \right\| \right) + \sum_{j \in N_i} S_1 \left(\left\| q_j - q_i \right\| \right) \left(p_j - p_i \right) + f_i^{leader} \left(q_i, p_i, q^{leader}, p^{leader} \right). \tag{38}$$

Заключение

Таким образом, с помощью предложенного метода синтезируется координирующее управление, обеспечивающее согласованное управление БПЛА в группе с целью достижения желаемой траектории движения. При решении задачи управления полетом группы БПЛА, центральное место занимает выбор математической модели для описания пространственного движения группы летательных аппаратов, поскольку уравнения динамики представляют собой сложную систему нелинейных дифференциальных уравнений, которая включает кинематические уравнения, уравнения сил, уравнения моментов, а также совокупность уравнений связей параметров движения в различных системах координат. Использованная в работе модель относительного движения позволяет декомпозировать совокупность уравнений динамики группы летательных аппаратов в набор моделей движения лидера и ведомых БПЛА.

Такой подход позволил упростить анализ задачи управления группой БПЛА, упростить переход к новой базовой системе отсчета при измерении координат относительного движения, а также упростить техническую реализацию выбранной базовой системы координат на борту БПЛА, что определяет простоту всей системы управления и в особенности её измерительной части.

В результате удалось реализовать принцип координирующего управления, которое обеспечивает перевод вектора переменных состояния в заданную область за один такт управления.

Intellectual Systems and Technologies

В группе агентов находится виртуальный лидер, который планирует маршрут движения группы в соответствии с заданной задачей и отслеживает определенную цель движения. Виртуальный лидер рассчитывает свое собственное управление с помощью алгоритма отслеживания траектории или отслеживания цели, чтобы двигаться по желаемой траектории. При этом сигнал наведения может позволить агентам собираться на позиции виртуального лидера и соответствовать вектору скорости виртуального лидера, чтобы обеспечить связь топологии связи в многоагентной системе и сформировать роевое формирование.

СПИСОК ЛИТЕРАТУРЫ

1. Ефанов В.Н., Мизин С.В., Неретина В.В. Управление полетом БПЛА в строю на основе координации взаимодействия группы летательных аппаратов // Вестник УГАТУ. 2014. Т. 18, № 1 (62). С. 114–121.

2. Pachter M., D'Azzo J.J., Proud A.W. Tight formation flight control // J. of Guidance, Control, and Dynamics, 2001, 24 (2): 246–254.

3. Zhang X.Y., Duan H.B., Yu Y.X. Receding horizon control for multi-UAVs close formation control based on differential evolution // Science China Information Sciences, 2010 (53): 223–235.

4. **Mu Y.** Research on aerodynamic coupling in UAV formation. Xi'an: Northwestern Polytechnical University, 2006.

5. Pollini L., Giulietti F., Innocenti M. Robustness to communication failures within formation flight // American Control Conference. Anchorage, AK, 2002: 2860–2866.

6. Gautier H., Simon L., Rachid A. Formation flight: Evaluation of autonomous configuration control algorithms // IEEE / RSJ Internat. Conf. on Intelligent Robots and Systems, San Diego, USA, 2007: 2628–2633.

7. Giulietti F., Pollini L., Innocenti M. Autonomous formation flight // IEEE Control Systems Magazine, 2000 (12): 566–572.

8. João S., Tunc S., Pravin V. Task planning and execution for UAV teams // IEEE Conference on Decision and Control. Atlantis, Bahamas, 2004: 3804–3810.

9. Lechevin N., Rabbath C.A., Lauzon M. Cooperative and deceptive planning of multi-formations of networked UCAVs in adversarial urban environments // AIAA Guidance, Navigation and Control Conference and Exhibit. Hilton Head, South Carolina, AIAA-2007-6410.

10. Reyna V.R., Pachter M., D'Azzo J.J. Formation flight control automation. AIAA: AIAA-94-3557, 1994.

11. **KOO T.J., Shahruz S.M.** Formation of a group of unmanned aerial vehicles (UAVs) // Proc. of the American Control Conf. Arlington, VA, 2001: 69–74.

12. Zong L., Xie F., Qin S. Intelligent optimal control of UAV formation flight based on MAS // J. of Aeronautics and Astronautics, 2008, 29 (5): 1326–1333.

13. Shin J., Kim H.J. Nonlinear model predictive formation flight // IEEE Transactions on Systems, Man and Cybernetics, 2009, 39 (5): 1116–1125.

14. Fidelis A.P.L., Tiauw H.G. A collision-free formation reconfiguration control approach for unmanned aerial vehicles // Internat. J. of Control, Automation, and Systems, 2010, 8 (5): 1100–1107.

15. Fidelis A.P.L., Tiauw H.G. Reconfiguration control with collision avoidance framework for unmanned aerial vehicles in three-dimensional space // J. of Aerospace Engineering, 2013, 26 (3): 637–645.

16. Xiong W., Chen Z., Zhou R. Optimization method for multi-aircraft formation reconstruction using hybrid genetic algorithm // J. of Aeronautics and Astronautics, 2008(29): 209–214.

17. Ye Q., Hu X., Ma H. Two-stage solution method for coordinated target assignment of multi-UAV formations // J. of Hefei University of Technology, 2015, 38 (10): 1431–1436.

18. **Dai J., Li X., Sun Y., et al.** Research on coordinated target allocation method for multi-formation ground attack // J. of System Simulation, 2009, 21 (8): 2148–2151.

19. **Reynolds C.W.** Flocks, herds, and schools: A distributed behavioral model // Comput. Graph. (ACM SIGGRAPH'87 Conf. Proc.), Jul. 1987, vol. 21, Pp. 25–34.

20. Czirók A., Vicsek M., Vicsek T. Collective motion of organisms in three dimensions // Physica A, 1999, 264 (1-2): 299–304.

21. Qiu H., Duan H. Pigeon interaction mode switch-based UAV distributed flocking control under obstacle environments // ISA Transactions, 2017, 71 (1): 93–109.

22. Jadbabaic A., Lin J., Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Transactions on Automatic Control, 2003, 48 (6): 988–1001.

23. Cucker F., Smale S. Emergent behavior in flocks // IEEE Trans Autom Control, 2007, 52 (5): 852–862.

24. **Ballerini M., Cabibbo N., Candelier R.** Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study // Proc. of the National Academy of Sciences, 2008, 105 (4): 1232–1237.

25. George M., Ghose D. Reducing convergence times of self-propelled swarms via modified nearest neighbor rules // Physic A, 2012, 391 (16): 4121–4127.

26. Vicsek T., Czirók A., Ben-Jacob E., et al. Novel type of phase transition in a system of self-driven particles // Physical Review Letters. 1995, 75 (6): 1226–1229.

27. Couzin D., Jens K., Richard J., Ruxton G.D., Franks N.R. Collective memory and spatial sorting in animal groups // J. of Theoretical Biology, 2002, 218 (1): 1–11.

28. Motsch S., Tadmor E. A new model for self-organized dynamics and its flocking behavior // J. of Statistical Physics, 2011, 144 (5): 923–947.

29. Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory // IEEE Transactions on Automatic Control, 2004, 51 (3): 401–420.

30. Li Z., Xue X. Cucher-Smale flocking under rooted leadership with fixed switching topologies // SIAM J. on Applied Mathematics, 2010, 70 (8): 3156–3174. DOI: 10.1137/100791774

REFERENCES

1. Yefanov V.N., Mizin S.V., Neretina V.V. Upravleniye poletom BPLA v stroyu na osnove koordinatsii vzaimodeystviya gruppy letatelnykh apparatov. *Vestnik UGATU*, 2014, vol. 18, No. 1 (62), Pp. 114–121. (rus)

2. Pachter M, D'Azzo J.J., Proud A.W. Tight formation flight control. *Journal of Guidance, Control, and Dynamics*, 2001, 24 (2): 246–254.

3. Zhang X.Y., Duan H.B., Yu Y.X. Receding horizon control for multi-UAVs close formation control based on differential evolution. *Science China Information Sciences*, 2010 (53): 223–235.

4. **Mu Y.** *Research on aerodynamic coupling in UAV formation*. Xi'an: Northwestern Polytechnical University, 2006.

5. Pollini L., Giulietti F., Innocenti M. Robustness to communication failures within formation flight. *American Control Conference*, Anchorage, AK, 2002: 2860–2866.

6. Gautier H., Simon L., Rachid A. Formation flight: Evaluation of autonomous configuration control algorithms. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA, 2007: 2628–2633.

7. Giulietti F., Pollini L., Innocenti M. Autonomous formation flight. *IEEE Control Systems Magazine*, 2000 (12): 566–572.

8. João S., Tunc S., Pravin V. Task planning and execution for UAV teams. *IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004: 3804–3810.

9. Lechevin N., Rabbath C.A., Lauzon M. Cooperative and deceptive planning of multi-formations of networked UCAVs in adversarial urban environments. *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, AIAA-2007-6410.

10. Reyna V.R., Pachter M., D'Azzo J.J. Formation flight control automation. AIAA: AIAA-94-3557, 1994.

11. **KOO T.J., Shahruz S.M.** Formation of a group of unmanned aerial vehicles (UAVs). *Proceedings of the American Control Conference*, Arlington, VA, 2001: 69–74.

12. Zong L., Xie F., Qin S. Intelligent optimal control of UAV formation flight based on MAS. *Journal of Aeronautics and Astronautics*, 2008, 29 (5): 1326–1333.

13. Shin J., Kim H.J. Nonlinear model predictive formation flight. *IEEE Transactions on Systems, Man and Cybernetics*, 2009, 39 (5): 1116–1125.

14. Fidelis A.P.L., Tiauw H.G. A collision-free formation reconfiguration control approach for unmanned aerial vehicles. *International Journal of Control, Automation, and Systems*, 2010, 8 (5): 1100–1107.

15. Fidelis A.P.L., Tiauw H.G. Reconfiguration control with collision avoidance framework for unmanned aerial vehicles in three-dimensional space. *Journal of Aerospace Engineering*, 2013, 26 (3): 637–645.

16. Xiong W., Chen Z., Zhou R. Optimization method for multi-aircraft formation reconstruction using hybrid genetic algorithm. *Journal of Aeronautics and Astronautics*, 2008 (29): 209–214.

17. Ye Q., Hu X., Ma H. Two-stage solution method for coordinated target assignment of multi-UAV formations. *Journal of Hefei University of Technology*, 2015, 38 (10): 1431–1436.

18. Dai J., Li X., Sun Y., et al. Research on coordinated target allocation method for multi-formation ground attack. *Journal of System Simulation*, 2009, 21 (8): 2148–2151.

19. Reynolds C.W. Flocks, herds, and schools: A distributed behavioral model. *Comput. Graph. (ACM SIGGRAPH'87 Conf. Proc.)*, Jul. 1987, vol. 21, Pp. 25–34.

20. Czirók A., Vicsek M., Vicsek T. Collective motion of organisms in three dimensions. *Physica A*, 1999, 264 (1-2): 299–304.

21. Qiu H., Duan H. Pigeon interaction mode switch-based UAV distributed flocking control under obstacle environments. *ISA Transactions*, 2017, 71 (1): 93–109.

22. Jadbabaic A., Lin J., Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 2003, 48 (6): 988–1001.

23. Cucker F., Smale S. Emergent behavior in flocks. IEEE Trans Autom Control, 2007, 52 (5): 852-862.

24. Ballerini M., Cabibbo N., Candelier R. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 2008, 105 (4): 1232–1237.

25. George M., Ghose D. Reducing convergence times of self-propelled swarms via modified nearest neighbor rules. *Physic A*, 2012, 391 (16): 4121–4127.

26. Vicsek T., Czirók A., Ben-Jacob E., et al. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 1995, 75 (6): 1226–1229.

27. Couzin.D., Jens K., Richard J., Ruxton G.D., Franks N.R. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 2002, 218 (1): 1–11.

28. Motsch S., Tadmor E. A new model for self-organized dynamics and its flocking behavior. *Journal of Statistical Physics*, 2011, 144 (5): 923–947.

29. Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions* on Automatic Control, 2004, 51 (3): 401–420.

30. Li Z., Xue X. Cucher-Smale flocking under rooted leadership with fixed switching topologies. *SIAM Journal on Applied Mathematics*, 2010, 70 (8): 3156–3174. DOI: 10.1137/100791774

INFORMATION ABOUT AUTHOR / СВЕДЕНИЯ ОБ АВТОРЕ

Чжу Юйцин Zhu Yuqing E-mail: 1918149382@qq.com

Поступила: 13.07.2022; Одобрена: 18.12.2022; Принята: 12.01.2023. Submitted: 13.07.2022; Approved: 18.12.2022; Accepted: 12.01.2023.
Research article DOI: https://doi.org/10.18721/JCSTCS.15403 UDC 004.032



TECHNIQUE FOR AUTOMATING CHARGING OF AN ELECTRIC VEHICLE BASED ON A RASPBERRY PI CONTROLLER USING NEURAL NETWORKS

N.A. Vlasenko¹, A.I. Dusaeva², I.V. Nikiforov³ ⊠ , D.S. Prelovskii⁴

^{1,2,3,4} Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

[™] igor.nikiforovv@gmail.com

Abstract. The expansion of Russian market of electric and autonomous vehicles leads to an increase in demand for automation of contactless charging (without driver participation). The article proposes a method of contactless charging of electric vehicles, which involves automatically determining the type of car charging connector, selecting the appropriate charger and connecting it to the charging connector of an electric vehicle through the use of a robot manipulator. A feature of the technique is the determination of the type and coordinates of the location of the charging connector of the car by reading images obtained from the camera of a gas station in real time and processing them with a convolutional neural network model. A study was conducted, and a function was selected that allows optimally solving the problems of classification of charging connectors, which ensures maximum accuracy of the result. The volume of the training sample for the neural network was used in the amount of 10,000 images from a synthetic data set, which was created on the basis of three types of the most popular threedimensional models of charging connectors on various backgrounds. The proposed technique is implemented in a prototype of a software and hardware control complex for a manipulative robot based on a Raspberry Pi controller.

Keywords: electric vehicle, computer vision, convolutional neural network, robotic arm, charging

Citation: Vlasenko N.A., Dusaeva A.I., Nikiforov I.V., Prelovskii D.S. Technique for automating charging of an electric vehicle based on a Raspberry Pi controller using neural networks. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 37–50. DOI: 10.18721/JCSTCS.15403

Интеллектуальные системы и технологии

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15403 УДК 004.032



МЕТОДИКА АВТОМАТИЗАЦИИ ЗАРЯДКИ ЭЛЕКТРОМОБИЛЯ НА БАЗЕ КОНТРОЛЛЕРА RASPBERRY PI С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Н.А. Власенко¹, А.И. Дусаева², И.В. Никифоров³ ⊠ , Д.С. Преловский⁴

^{1,2,3,4} Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация

^{III} igor.nikiforovv@gmail.com

Аннотация. Расширение российского рынка электрических и автономных транспортных средств ведет к увеличению спроса на автоматизацию бесконтактной зарядки (без участия водителя). В статье предложена методика бесконтактной зарядки электромобилей, которая предполагает автоматическое определение типа зарядного коннектора автомобиля, выбор соответствующего зарядного устройства и его подключение в зарядный коннектор электромобиля благодаря использованию робота-манипулятора. Особенностью методики является определение типа и координат расположения зарядного коннектора автомобиля за счет считывания изображений, получаемых с камеры автозаправочной станции в режиме реального времени и обработанных при помощи модели свёрточной нейронной сети. Проведено исследование и выбрана функция, позволяющая оптимально решать задачи классификации зарядных коннекторов, обеспечивающая максимальную точность результата. Объём обучающей выборки для нейронной сети использован в размере 10 000 изображений из синтетического набора данных, созданного на основе трёх типов наиболее популярных трёхмерных моделей зарядных коннекторов на различных фонах, приближенных к реальным условиям использования роботов, обслуживающих зарядные станции. Предложенная методика реализована в прототипе программно-аппаратного комплекса управления манипуляционным роботом на основе контроллера Raspberry Pi.

Ключевые слова: автоматизация, компьютерное зрение, свёрточная нейронная сеть, робот-манипулятор, зарядное устройство

Для цитирования: Vlasenko N.A., Dusaeva A.I., Nikiforov I.V., Prelovskii D.S. Technique for automating charging of an electric vehicle based on a Raspberry Pi controller using neural networks // Computing, Telecommunications and Control. 2022. T. 15, № 4. C. 37–50. DOI: 10.18721/ JCSTCS.15403

Introduction

To date, one of the most acute environmental problems in developed countries is the pollution of the atmosphere by exhaust gases from motor vehicles. The total level of environmental pollution by exhaust gases from total emissions of harmful substances in Europe is 72.9 % [1]. One of the possible ways to solve this problem is the widespread use of electric cars since they are more environmentally friendly than cars with an internal combustion engine and contribute to reducing the consumption of fossil fuels. This trend leads to the need to build a modern infrastructure for the vehicles maintenance [2].

Often, the construction of infrastructure for vehicles is carried out using robotic tools and systems, which makes it possible to automate processes and minimize human involvement [3]. One of the parts of such infrastructure is a contactless charging system with the use of robotic manipulators, which should provide a high-quality, reliable, and fast connection of the charging station with the charging socket

of the car. Accordingly, with the increase of electric vehicles in the Russian market [4, 5], the need for automatic contactless charging systems using robotic manipulators also increases, which is why *the relevance* of the chosen topic is determined.

In addition, the situation in Russia differs from other electric vehicle markets [4], since there are cars in our country that meet both European and Japanese standards. Guided by these considerations, the paper considers the most popular charging sockets of electric vehicles in the territory of the Russian Federation [5].

Research

Since the issue of the introduction and creation of automatic charging stations is relevant almost for the whole world, many research groups are developing automated charging systems for electric vehicles.

For example, M. Bell, J. Duro, T. Flynt et al. in their article [6] consider a robot manipulator for refuelling electric cars. Their research is aimed at developing a navigation system for the robot using the LiDar system, as well as a system of movement towards the charging connector option.

Researcher E.H.C Harik in the article [7] considers the possibility of introducing autonomous electric tractors into agriculture and proposes an autonomous charging station system that uses visual navigation and detection to connect a power cable to an outlet. In the article, E.H.C Harik uses only one type of charging connectors, which is a generalized model and is not used in real conditions.

A group of authors, B. Walzel, C. Sturm, J. Fabian, and M. Hirz, in their work [8] give a brief overview of both existing charging systems from large companies such as Tesla, Volkswagen, and various research projects, for example, a charging system from the Technical University of Dortmund. In addition, the article highlights the main problems associated with the development of such an automatic system: the location of the car in the parking space at the time of connecting the charger, various types of connectors. They also proposed the concept of an automated charging system.

The works considered in the study provide a solution to their task, but they also have several disadvantages associated with a low degree of automation, low connection quality (below 70 %), which leaves room for improvement of the car charging automation. Some solutions describe a generalized model and are not applied in real conditions.

The article suggests a different approach for charging electric vehicles with a robot manipulator. Its main task is to automatically determine the type of charging connector of the car and build the optimal trajectory from the charging station to the socket of the car using a robot manipulator. Also, a distinctive feature is the creation, training and testing of a neural network for classifying various charging sockets on a physical model of an automated charging system with the function of visual navigation of a robot manipulator.

The aim of the work is to develop a technique that provides contactless charging of electric vehicles by means of implementing a software and hardware complex based on the use of neural networks and a Raspberry Pi controller.

The methodology consists of three main steps presented below.

Step 1. Automatic detection of the type of car charging connector by reading the image from the video camera and its recognition by means of a neural network.

Step 2. Determining the coordinates of the charging socket and calculating the optimal trajectory for moving the selected charger to the socket of the electric vehicle.

Step 3. Connecting the charger to the charging connector of the electric vehicle by using a robot manipulator.

A feature of the technique is the determination of the type and coordinates of the location of the charging connector of the car by reading images obtained from the camera of a gas station in real time and processing them using a convolutional neural network model. The technique proposed in the paper is implemented in a software tool based on the Raspberry Pi 4 controller.

Finite automaton behaviour of the robot manipulator

The technique proposed in the paper was brought to implementation in a hardware and software complex, the basis of which is the behaviour described by a finite automaton. Fig. 1 shows a model of its behaviour in the form of a Mile machine [9], consisting of a finite number of states. Based on certain input data, there is a transition from one state to another and data transfer.

In a finite state machine, the following basic states can be distinguished:

- "take photo" - the state in which the camera located at the gas station takes a picture of the charging socket of an electric vehicle;

 "recognize socket" – the state in which the type of car charger is recognized by the image created by the camera;

 "change charger" – the state in which the robot manipulator changes the active charger to a suitable one for the current car;

 "calculate coordinates" and "calculate angle" – states in which, based on photos of the charging socket of an electric vehicle, the coordinates of its location and angle are calculated, respectively;

"plug" – connection of the charger by the robot manipulator to the charging socket of the electric vehicle;

– "charge" – a condition that characterizes the process of direct physical charging of an electric vehicle;

- "idle" - the state of the end of the charging process of the electric vehicle.

The most interesting states that will be considered in the paper are: "recognize socket", "calculate coordinates" and "calculate angle".

Automation of determining the type of car charging connector

In the "recognize socket" state, the type of car charging socket is determined based on image recognition [10].

There are a large number of charging sockets for electric cars. They differ in charging speed. According to statistics [4] the most popular cars for 2020–2021 are Nissan Leaf, Porsche Taycan, Audi e-tron, Tesla Model 3, Mitsubishi i-MiEV and others. These models use the following types of charging connectors:



Fig. 1. Behavior model of robotic arm system



Fig. 2. Models of charging sockets in use

- standard connector of the first type (SAE J1772, J-PLUG, J-plug), shown in Fig. 2*a* is a North American standard, common among American and Japanese vehicles. It is single-phase and includes 5 contacts;

- connector of the second type (Mennekes), shown in Fig. 2b is a European standard with 7 contacts, at the moment it is the most common among electric vehicles in Europe and China, can be single-phase or three-phase;

combined type (CCS Combo 2), shown in Fig. 2c is an improved version of the second type with 9 contacts. The second type of the cable can also be used for this connector.

Thus, the work selected the above tips for recognition automation as the most common in Russia. Their example shows the applicability and effectiveness of the methodology. At the same time, the applicability of the technique is not limited to the considered types of electric vehicles and their charging sockets. It is universal and can be used, among other things, for other electric vehicles. To do this, it will be necessary to expand the data set for training the neural network with images with additional photos of charging connectors and retrain the neural network.

Figure 3 shows the architecture of a neural network for determining the type of electric car charger, which consists of four convolution layers [11] (Conv2D with the number of neurons 8, 12, 16 and 18, respectively), performing the operation of multiplying the image matrix (28×28) by the convolution core matrix (3×3). Convolution layers alternate with four layers of subdiscretization (MaxPooling2D), which are necessary to compress the size of the extracted image feature map [12]. The work uses a convolutional neural network, because this architecture has proven itself well in image recognition tasks, and the number of layers is justified by the highest efficiency and quality of recognition in comparison with other numbers of layers.

Consider the layers of a neural network. The thinning layer (Dropout) is used to solve the problem of retraining the network, converting to a one-dimensional vector (Flatten). Next is a fully connected layer with a linear activation function (Dense 128 + ReLU), a thinning layer (Dropout) and the last fully connected layer with a sigmoid activation function (Dense 4 + sigmoid), which determines independent probabilities for determining the coordinates of the object in the frame. At the output of the neural network, we get a two-dimensional array array [samples] [4], where samples is the number of images submitted to the input of the neural network, and 4 is the number of coordinates received.

To classify the connectors, the Softmax activation function was used, which determines the dependent probabilities of distribution over three possible classes for classifying objects. At the output of the neural network, we get a two-dimensional array array [samples] [3], where samples is the number of images submitted to the input of the neural network, and 3 is the probability distribution by class.

To train a neural network that implements the recognition mechanism of the charging socket of an electric vehicle, a data set [13] from CAD models (Computer-Aided Design) was prepared.



Fig. 3. Architecture of classification charging sockets neural network

The approach used in the work makes it possible to obtain many images of the required objects in various positions, lighting and environment, and automatically mark up the data.

10,000 images were obtained in the work. In the Blender program, a program was created using the bpy library, where a random angle and position of the part on the scene are calculated within acceptable limits, since it should be facing the camera and should not go beyond the stage. After that, the background is applied, the image is saved with the settings set.

Choosing an optimizer for classification of charging connectors

The study and selection of the best optimizer for the task of classifying charging connectors was carried out. These optimizers are used in most machine-learning tasks.

Stochastic gradient descent (SGD) [14] is one of the simplest methods of minimizing the loss function, on the basis of which other optimizers are built:

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla_{\theta} J(\theta_t),$$

where θ – network parameters (weights), η – learning rate, $J(\theta_t)$ – loss function.

In the connector classification problem, this optimizer is at $\eta = 0.001$ (this value is optimal, because with an increase in this hyperparameter, the final classification accuracy will decrease due to a large number of local minima omissions, convergence may not be achieved, with a decrease, the convergence rate will decrease significantly, especially when entering the plateau zone), it has the lowest convergence, the classification accuracy on the test set was 35.64 %. This problem arises due to the impossibility of adaptive changes in the learning rate and, accordingly, the step length.

The Root Mean Square Propagation Algorithm (RMS Pro) is used for batch optimization, in which data is processed in blocks. Its main idea is to preserve the moving average of the squares of the gradients for each weight. The learning rate is adapted by dividing it by an exponentially decreasing average of the squares of the gradients:

$$v_{t} = \beta v_{t-1} + (1-\beta) \left(\nabla_{\theta} J(\theta_{t}) \right)^{2},$$

$$\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{\nu_t} + \epsilon} \nabla_{\theta} J(\theta_t),$$

where v_t – moving average, β – accumulation coefficient, ϵ – smoothing parameter.

In this case, the loss function for the validation set and the classification accuracy are unstable, the accuracy on the test set was 81.54 %.

Adaptive Moment Estimation (Adam) is an advanced RMS Pro algorithm using momentum and one of the most effective and most frequently used optimizers in machine learning [15]. It calculates the adaptive learning rate for each parameter with a slight change in weights for characteristic features. To ensure this, it is necessary to preserve the exponentially decreasing average of the squares of the gradients in previous iterations and the exponentially decreasing average of the past gradients. Instead of using the entire dataset to calculate the actual gradient, this optimization algorithm uses a randomly selected subset of the data to create a stochastic approximation:

$$m_{t} = \beta_{1}m_{t-1} + (1 - \beta_{1})\nabla_{\theta}J(\theta_{t})$$

$$v_{t} = \beta_{2}v_{t-1} + (1 - \beta_{2})(\nabla_{\theta}J(\theta_{t}))^{2}$$

$$\hat{m}_{t} = \frac{m_{t}}{1 - \beta_{1}^{t}}, \quad \hat{v}_{t} = \frac{v_{t}}{1 - \beta_{1}^{t}}$$

$$\theta_{t+1} \leftarrow \theta_{t} - \frac{\eta}{\sqrt{\hat{v}_{t}} + \epsilon}\hat{m}_{t} = \theta_{t} - \frac{\eta}{\sqrt{\hat{v}_{t}} + \epsilon}\left(\beta_{1}\hat{m}_{t-1} + \frac{1 - \beta_{1}}{1 - \beta_{1}^{t}}\frac{\Delta loss}{\Delta\theta_{t}}\right)$$

where $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$; $0 \le \beta_1 \le 1.0 \le \beta_2 \le 1 - \text{accumulation coefficient}$; $v_t - \text{average non-centered variance}; m_t - \text{exponential moving average}.$

At the end of the training, the classification accuracy, and the loss function for the training set and for the validation set begin to diverge, which entails retraining, so the accuracy with this optimizer was 84.15 %.

The algorithm of adaptive estimation of moments with infinite norms (Adamex) uses the inertial moment of gradient distribution:

$$v_{t} = \beta_{2}^{k} v_{t-1} + (1 - \beta_{2}^{k}) |\nabla_{\theta} J(\theta_{t})|^{k}$$
$$u_{t} = \sqrt[\infty]{v_{t}} = \max\left(\beta_{2}^{t-1} |\nabla_{\theta} J(\theta_{1})|, \beta_{2}^{t-2} |\nabla_{\theta} J(\theta_{2})|, \dots, \beta_{2} |\nabla_{\theta} J(\theta_{t-1})|, |\nabla_{\theta} J(\theta_{t})|\right)$$
$$\theta_{t+1} \leftarrow \theta_{t} - \frac{\eta}{u_{t}} \hat{m}_{t}$$

$$\eta = 0.002, \beta_1 = 0.9, \beta_2 = 0.999.$$

At the beginning of the training, there is a big difference and instability of the results of functions on the test and validation datasets, however, by the hundredth epoch, the results gradually converge; nevertheless, the recognition accuracy of the test set was 77.23 %.

Adaptive estimation of moments with Nesterov acceleration (NAdam) is an improved Adam algorithm. In this optimizer the next position of the exponential running average (\hat{m}_t) is not predicted in advance, and the gradient update formula compared to the Adam algorithm changes accordingly as follows:

$$\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\beta_1 \hat{m}_t + \frac{1 - \beta_1}{1 - \beta_1^t} \frac{loss}{\theta_t} \right),$$

$$\eta = 0.002, \quad \beta_1 = 0.9, \quad \beta_2 = 0.999, \quad \epsilon = 10^{-7}.$$

Thanks to this algorithm, significant success was achieved: instability of functions was eliminated, the results of their work converged, the recognition accuracy of the test set was 91.59 %.

Table 1 shows the final comparison of the accuracy of the classification of optimizers for each charging connector. The rows of the table are the tests performed to determine accuracy by class (Type 1, Type 2, Type 3), the total accuracy on training data (Train_accuracy) and on test data (Test_accuracy), and the columns are the optimization functions under consideration. The cells show the results of calculating the recognition accuracy on the test data set.

Table 1

	Adam	NAdam	RMS Pro	Adamax	SGD
Type 1	0.7554	0.8785	0.6677	0.7862	0.0615
Type 2	0.9031	0.8985	0.8062	0.8367	0.9978
Туре 3	0.8661	0.9708	0.9723	0.7228	0.0077
Train_accuracy	0.8583	0.8938	0.8982	0.7632	0.5408
Test_accuracy	0.8415	0.9159	0.8154	0.7723	0.3564

The resulting classification accuracy for different optimizers

Automation of determining the trajectory of the robot manipulator from the charger to the charging connector of the car

In the "calculate coordinates" and "calculate angle" states (see Fig. 1), the trajectory of the robot manipulator [16] with the selected active charger to the charging socket of the electric vehicle is determined by using images from the camera.

One of the main physical characteristics of the camera, which takes pictures of the charging connector of the car at the gas station, are its viewing angles, which are calculated by the formulas:

$$\alpha = 2 \arctan\left(\frac{h}{2f}\right),$$
$$\beta = 2 \arctan\left(\frac{v}{2f}\right),$$

where α , $\beta \in (0;360)$, rad; *h*, *v* – sensor dimensions, mm; *f* – focal length, mm.

To create an algorithm for the movement of the robot manipulator from the charger to the charging connector of the car, it is necessary to enter the definitions listed below.

Definition 1. F_x , $F_y \in Q | 0 < F_x < 0.5\alpha \land 0 < F_y < 0.5\beta$ we will call the final shifts, the relative values by which the object in the image is shifted relative to the optical axis of the camera. Let $x, y \in (0,1)$, then:

$$F_x^{k+1} = \left(x_0^k + \frac{x_1^k - x_0^k}{2}\right) * \alpha, \quad F_y^{k+1} = \left(y_1^k + \frac{y_1^k - y_0^k}{2}\right) * \beta,$$

where α , β – camera viewing angles; x_0, x_1 – diagonal coordinates along the X-axis, the fraction relative to the total width of the image; y_0, y_1 – diagonal coordinates along the Y-axis, the fraction relative to

the total height of the image. *Definition 2.* γ_x^{k+1} , $\gamma_y^{k+1} \in Q$ we will call the rotation angles (rad), the values by which the servomotor will rotate at each new iteration, based on its current position and the coordinates of the object in the resulting image:

$$\gamma_x^{k+1} = \left(\gamma_x^k \pm F_x^k\right), \quad \gamma_y^{k+1} = \left(\gamma_y^k \pm F_y^k\right),$$

where F_x , F_y – final shifts. Thus, we get:

$$\gamma^{k+1} = \left(\gamma^k \pm \left(x_0^k + \frac{x_1^k - x_0^k}{2}\right) * \alpha\right).$$

The algorithm for calculating the trajectory of movement can be represented as a sequence of steps. Step 1. Take the initial values h, v – sensor dimensions, f – focal length, x – coordinate vector.

Step 2. Accept
$$\alpha := 2 \arctan\left(\frac{h}{2f}\right), \ \beta := 2 \arctan\left(\frac{v}{2f}\right), \ k = \overline{0, n}.$$

Step 3. Accept $F_x^{k+1} := \left(x_0^k + \frac{x_1^k - x_0^k}{2}\right) * \alpha.$

Step 4. If $F_x^{k+1} < 0.5$, accept $\gamma_x^{k+1} := (\gamma_x^k - F_x^k)$, if $F_x^{k+1} > 0.5$, accept $\gamma_x^{k+1} := (\gamma_x^k + F_x^k)$ otherwise $\gamma_x^{k+1} := (\gamma_x^k).$

Step 5. Return γ_{x}^{k+1} , accept k = k+1.

In this case, the local minimum of the function is achieved when the optical axis of the camera and the calculated center of the recognized object coincide, since in the interval between iterations there is a possibility of changing the position of the tracked object or changing the selection results in case the object does not fully enter the frame during the first iteration. The algorithm is carried out before the condition of changing axes.

Tools and technologies for implementing the methodology

The set of tools and technologies for the implementation of each individual module of the software product includes:

- to write the motor control module for the Arduino Uno board, an ATmega328P-based microcontroller, the Arduino IDE 1.8.19 development environment and the C++ programming language with the Wiring framework and an extensive set of libraries were used;

- the motion vector calculation module for the Raspberry Pi 4 B board (Cortex-A72 [17]) was developed using Python 3.6;

 training and testing of neural networks was conducted in the Jupiter Notebook 6.4.8 interactive development environment using open libraries for Tensorflow 2.2.0 and Keras 2.2.1 machine learning and Python 3.6 programming language;



Fig. 4. Software module diagram that implements the proposed technique

 writing a program for rendering images and creating a data set for processing by neural networks was carried out on the Ubuntu 21.04 operating system in open-source 3D modelling software Blender 3.0.1 in Python.

In accordance with the stages of the methodology proposed in the work, Fig. 4 shows a diagram of the modules of the software tool. It consists of 5 main modules.

The "Rendering" module is necessary to compile a data set for training a neural network: the components "Socket_Type1", "Socket_Type2", "Socket_Type3" start the process of creating images with three types of charging connectors on different backgrounds.

The module "Jupyter Notebook" trains neural networks to determine the coordinates and type of the charging connector of the car: the component "CNN_coordinates" implements the definition of the coordinates of the required object in the frame; the component "CNN_class" implements the definition of the connector class.

The Raspberry module is the main one, it starts the initialization of the camera and the process of creating photos in real time loading neural network models from the Jupyter Notebook module, basic calculations and data transfer to the Adruino module: the "Preparation" component prepares directories, checks the existence of the necessary directories, creates them if necessary; the "Camera" component starts the process of creating a photo; the "Model" component processes the received photo using a neural network model; the "Calculate" component performs basic calculations with the received data; the "Communication" component establishes a connection to the Arduino Uno board, sends a command to the ttxAXMx port.

The "Arduino module" controls the motors, changing the position of the manipulator in space.

The "Dataset" module stores data sets for training, validation, and testing of neural networks.

Testing of software and hardware

The prototype of the gas station robot manipulator created in the work, shown in Fig. 5, meets the following requirements:

– recognizes the type of charging socket of the vehicle with an accuracy of at least 80 % before instructing the robot arm to grab the charger and start moving it to the charging socket of the vehicle (average recognition accuracy of 91.59 %);

- determines the coordinates of the car's charging socket and selects the optimal trajectory;

- starts moving when the coordinates of the object are received;



Fig. 5. Prototype of robotic arm

- at each iteration, searches for the center of the selected object for subsequent calculations of the rotation angles of the motors of which it consists.

With the number of images in the training dataset equal to 1500 images and an acceptable error of p = 0.01 and p = 0.05 45 % and 69 % of images are recognized correctly, respectively, while with an error of p = 0.2 % - 96 %. When increasing the size of the input dataset to 10,000 images and with an acceptable error of p = 0.01, 96.41 % of the images are recognized correctly.

Table 2 shows a comparison of the hardware boards that can be used in robotic arm for neural networks processing. Raspberry Pi 4 B occupies a leading position in image processing for machine learning [18] and image compression [19]. It provides fast enough speed of data processing and image recognition in order to produce output in a reasonable amount of time. So Raspberry Pi 4 B was used to build a robotic arm model.

At the same time, Arduino Uno controller was chosen for conversion of calculated coordinates into rotational output (to servomotors), since it has good reputation in the similar tasks solutions [20].

Fig. 6 shows an example of recognition and allocation of charging connectors, as well as accuracy for each type.

Conclusion

The paper proposes a method of contactless charging of electric vehicles, which involves automatically determining the type of car charging connector, selecting the appropriate charger, and connecting it to the charging connector of an electric vehicle using a robot manipulator.

A prototype robot manipulator was developed that implements the proposed technique and has the following properties:

– determines the type of vehicle charging connector with an accuracy of 91.59 % due to the use of a convolutional neural network. The data collection method used to train a neural network based on 3D models of charging sockets allows you to create data sets in different areas (at gas stations, in urban areas, parking lots, on federal highways, etc.) with different natural and weather conditions that affect lighting and visibility (rain, fog, blizzard, night time) at a lower cost;

- determines the coordinates for the movement of the robot manipulator with an accuracy of 96.41 % due to the use of the optimal loss minimization function.

Based on wide and deep comprehensive analysis Raspberry Pi 4 B board was selected as the most powerful and fast enough board to handle and process input data and provide relevant output in a reasonable amount of time.



Fig. 6. Demonstration of neural networks output

Table 2

Comparison of hardware boards for data processing in robotic arm

ard name	CPU	GPU	RAM (GB)	Memory bandwidth (MBps)		ine Learning lages/sec)	Compression bixels/sec)
Bo				Read	Write	Mach (in	Image (M
Raspberry Pi 4 B	1.5GHz Quad core Cortex-A72 (ARM v8) 64-bit	VideoCore VI OpenGL ES 3.x	8	4129	4427	7.30	33.2
Orange PI 4	Rockchip RK3399(Cortex- A72+Cortex-A53) 1.8 GHz	Mali-T860	4	2972	3106	3.80	13.4
LeMaker Banana Pi	ARM Allwinner @ 960 MHz	ARM Mali-400 MP2 GPU dual-core	1	805	1358	0.30	2.54
BeagleBone AI-64	Texas Instruments Jacinto TDA4VM 2x ARM Cortex-A72 2 GHz	PowerVR® Rogue™ 8XE GE8430 3D	4	3679	3843	5.59	17.0

The developed prototype of the software and hardware solution allows reducing human involvement in the process of automating the charging of vehicles.

REFERENCES

1. Gómez-Vilchez J.J., Julea A., Peduzzi E., Pisoni E., Krause J., Siskos P., Thiel C. Modelling the impacts of EU countries' electric car deployment plans on atmospheric emissions and concentrations. *European Transport Research Review*, 2019, vol. 11, No. 40, 1 p. DOI: 10.1186/s12544-019-0377-1

2. Zhang H., Sheppard C., Lipman T., Zeng T., Moura S. Charging infrastructure demands of shareduse autonomous electric vehicles in urban areas. *Transportation Research Part D Transport and Environment*, 2020, No.78, 102210 p. DOI: 10.1016/j.trd.2019.102210 3. Sanatov D.V. et al. *Electric vehicle market and charging infrastructure development perspectives in Russia: Expertise and analysis report.* POLYTECH-PRESS, 2021, 44 p.

4. Labudin A.V., Eshenkova N.S., Burdukova A.O. Current trends and prospects for the development of the electric vehicle market in Russia. *Economics and Management of the National Economy (Saint Petersburg)*, 2021, No. 14 (16), 143 p.

5. **Komarova M.V.** Analiz rynka elektromobiley v Rossii. *Innovatsii. Nauka. Obrazovaniye*, 2020, No. 21, Pp. 276–281. (rus)

6. Bell M., Duro J., Flynt T., Hameed I., Lang G., Park F. Autonomous electric vehicle charging system. *Systems and Information Engineering Design Symposium (SIEDS)*, 2019, Pp. 1–6. DOI: 10.1109/SIEDS.2019.8735620

7. Harik E.H.C. Design and implementation of an autonomous charging station for agricultural electrical vehicles. *Applied Sciences*, 2021, vol. 11, No. 13: 6168. DOI: 10.3390/app11136168

8. Walzel B., Sturm C., Fabian J., Hirz M. Automated robot-based charging system for electric vehicles. *International Stuttgarter Symposium*, 2016. DOI: 10.1007/978-3-658-13255-2_70

9. Salle D., Herrero C.H., Outon J., Esnaola U., Lopez-de-Ipiña K. State machine based architecture to increase flexibility of dual-arm robot programming. *Bioinspired Computation in Artificial Systems*, 2015, No. 2, 98 p. DOI: 10.1007/978-3-319-18833-1_11

10. Al-Saffar A.M., Tao H., Talab M.A. Review of deep convolution neural network in image classification. *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2017, Pp. 26-31. DOI: 10.1109/ICRAMET.2017.8253139

11. Cuiping S., Tan C., Wang T., Wang L. A waste classification method based on a multilayer hybrid convolution neural network. *Applied Sciences*, 2021, vol. 11, No. 18: 8572. DOI: 10.3390/app11188572

12. Parasich A.V., Parasich V.A., Parasich I.V. Formirovaniye obuchayushchey vyborki v zadachakh mashinnogo obucheniya. *Informatsionno-Upravlyayushchiye Sistemy*, 2021, No. 4, Pp. 61–70. (rus). DOI: 10.31799/1684-8853-2021-4-61-70

13. Voinov N., Drobintsev P., Kotlyarov V., Nikiforov I. Distributed OAIS-Based digital preservation system with HDFS technology. *Proceedings of the 20th Conference of Open Innovations Association FRUCT*, St. Petersburg, 2017, Pp. 491–497. DOI: 10.23919/FRUCT.2017.8071353

14. Chernorutskiy I., Drobintsev P., Kotlyarov V., Voinov N. A new approach to generation and analysis of gradient methods based on relaxation function. 2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim), Cambridge, UK, 2017, Pp. 83–88. DOI: 10.1109/UKSim.2017.14

15. **Postalcioglu S.** Performance analysis of different optimizers for deep learning based image recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2019, vol. 34, No. 2. DOI: 10.1142/S0218001420510039

16. **Rostova E.N., Rostov N.V., Yan Z.** Neural network compensation of dynamic errors in a position control system of a robot manipulator. *Computing, Telecommunications and Control*, 2020, vol. 13, No. 1, Pp. 53–64. DOI: 10.18721/JCSTCS.13105

17. **Pankov P.A., Nikiforov I.V., Drobintsev D.F.** Hardware and software data processing system for research and scientific purposes based on Raspberry Pi 3 microcomputer. *Proceedings of the Institute for System Programming of the RAS*, 2020, vol. 32, No. 3, Pp. 57–70. DOI: 10.15514/ISPRAS-2020-32(3)-5

18. Ozkan Z., Bayhan E., Namdar M., Basgumus A. Object detection and recognition of unmanned aerial vehicles using Raspberry Pi platform. 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, Pp. 467–472. DOI: 10.1109/ISMSIT52890.2021.9604698

19. Sahitya S., Lokesha H., Sudha L.K. Real time application of Raspberry Pi in compression of images. *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology* (*RTEICT*), Bangalore, India, 2016, Pp. 1047–1050, DOI: 10.1109/RTEICT.2016.7807990

20. Oza V., Mehta P. Arduino robotic hand: Survey paper. *International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, India, 2018, Pp. 1–5, DOI: 10.1109/ICSCET.2018.8537312

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Власенко Наталия Андреевна Nataliia A. Vlasenko E-mail: nt.vlasenko@yandex.ru

Дусаева Анеля Ильясовна Anelya I. Dusaeva E-mail: an.dusaeva@gmail.com

Никифоров Игорь Валерьевич Igor V. Nikiforov E-mail: igor.nikiforovv@gmail.com

Преловский Дмитрий Сергеевич Dmitrii S. Prelovskii E-mail: dimaprelovski@gmail.com

Поступила: 29.11.2022; Одобрена: 17.12.2022; Принята: 12.01.2023. Submitted: 29.11.2022; Approved: 17.12.2022; Accepted: 12.01.2023. Научная статья DOI: https://doi.org/10.18721/JCSTCS.15404 УДК 004.312.44



ИССЛЕДОВАНИЕ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ПРОГРАММНОЙ И АППАРАТНЫХ РЕАЛИЗАЦИЙ ОПЕРАЦИИ СУММИРОВАНИЯ ТРАНСПОНИРОВАННЫХ МАТРИЦ

А.П. Антонов¹ [№] , Д.С. Беседин², А.С. Филиппов³

^{1,2,3} Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация

[™] antonov@eda-lab.ftk.spbstu.ru

Аннотация. Статья посвящена исследованию и сравнительному анализу программной и аппаратной реализации операции суммирования транспонированных матриц и её модифицированного варианта: операции транспонирования суммы матриц. Особенностью исследования является использование для получения аппаратной реализации средств высокоуровневого синтеза. Актуальность исследования обусловлена широким использованием матричных операций для решения задач различных классов, степенной асимптотической сложностью матричных вычислений и отсутствием данных об использовании данного инструментария в задачах создания аппаратных устройств для матричных вычислений. Предложен пошаговый метод синтеза и оптимизации аппаратного устройства. Проведено сравнительное исследование программных и аппаратных реализаций двух вычислительных задач. Показано, что большой выигрыш производительности аппаратных реализаций получается за счет увеличения степени параллелизма вычислений. Дополнительно сделаны выводы о неэффективности попыток достичь высоких тактовых частот, а также об увеличении затрачиваемых ресурсов при увеличении быстродействия за счет распараллеливания.

Ключевые слова: аппаратная реализация, производительность, аппаратные затраты, FPGA, параллельные вычисления, конвейеризация

Финансирование: Работа выполнена при поддержке Госзадания СПбПУ FSEG-2022-0001.

Для цитирования: Антонов А.П., Беседин Д.С., Филиппов А.С. Исследование и сравнительный анализ эффективности программной и аппаратных реализаций операции суммирования транспонированных матриц // Computing, Telecommunications and Control. 2022. T. 15, № 4. C. 51–63. DOI: 10.18721/JCSTCS.15404 Intellectual Systems and Technologies

Research article DOI: https://doi.org/10.18721/JCSTCS.15404 UDC 004.312.44



RESEARCH AND COMPARATIVE ANALYSIS OF THE EFFECTIVENESS OF SOFTWARE AND HARDWARE IMPLEMENTATIONS OF THE OPERATION OF SUMMING TRANSPOSED MATRICES

A.P. Antonov¹ \bowtie , D.S. Besedin², A.S. Filippov³

^{1,2,3} Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation ⊠ antonov@eda-lab.ftk.spbstu.ru

Abstract. The article is devoted to the study and comparative analysis of the software and hardware implementation of the operation of summing transposed matrices and its modified version – the operation of transposing the sum of matrices. A feature of the study is the use of high-level synthesis tools to obtain a hardware implementation. The relevance of the study is due to the widespread use of matrix operations for solving problems of various classes, the power asymptotic complexity of matrix calculations and the lack of data on the use of this toolkit in the tasks of creating hardware devices for matrix calculations. A step-by-step method of synthesis and optimization of a hardware device is proposed. A comparative study of software and hardware implementations of two computational tasks is carried out. It is shown that a large gain in the performance of hardware implementations are drawn about the inefficiency of attempts to achieve high clock frequencies, as well as about the increase in resources spent with increased speed due to parallelization.

Keywords: hardware implementation, performance, hardware costs, FPGA, parallel computing, pipelining

Funding: The work was carried out with the support of the Goszadanie of SPbPU FSEG-2022-0001.

Citation: Antonov A.P., Besedin D.S., Filippov A.S. Research and comparative analysis of the effectiveness of software and hardware implementations of the operation of summing transposed matrices. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 51–63. DOI: 10.18721/JCSTCS.15404

Введение

Матричные вычисления широко используются для решения задач различных классов. При этом асимптотическая сложность матричных вычислений имеет степенную зависимость от числа строк/столбцов обрабатываемых матриц, например, в общем случае: $O(n^3) - для$ умножения матриц, $O(n^2) - для$ сложения матриц. Поэтому задача повышения производительности вычислительных систем при выполнении базовых матричных операций актуальна [1, 2].

Матричные вычисления — это тип вычислений, позволяющий осуществлять пространственное распараллеливание вычислений, что является следствием того, что вычисления сводятся к операциям с независимыми, с точки зрения вычислительного процесса, элементами исходных матриц. Кроме того, отсутствие зависимости по данным для большинства базовых матричных операций позволяет использовать конвейеризацию (распараллеливание во времени), обеспечивающую параллельное (одновременное) выполнение операций считывания элементов матриц, арифметических действий над ними и записи результата одновременно для *P* элементов результирующей матрицы. Очевидно, что теоретически пространственное распараллеливание вычислений может дать кратный степени распараллеливания рост производительности. Соответственно, конвейеризация – кратный коэффициенту конвейеризации, т. е. числу реализованных этапов конвейера. Однако такие теоретические оценки являются излишне оптимистичными, поскольку не учитывают ограничения, связанные с пропускной способностью подсистем, обеспечивающих считывание/запись данных, т. е. считывание и запись элементов матриц, а также с доступными логическими ресурсами для реализации арифметических операций над элементами матриц [3].

Средства и подходы, традиционно используемые для повышения производительности вычислений, основанных на матричных операциях, хорошо известны. Это и многопоточное, многоядерное и/или многопроцессорное пространственное распараллеливание, реализуемое на традиционных процессорах, имеющих N физических ядер и позволяющих реализовать до N*2 потоков. Это и графические карты, работающие в режиме вычислителя, в англоязычной литературе называемом General Purpose Graphic Processing Unit (GPGPU), и имеющие систолическую SIMD (Single Instruction Multiple Data) — одна инструкция много данных — архитектуру. Это и FPGA (Field Programmable Gate Arrays) — СБИС (Сверхбольшие Интегральные Схемы) с внутренней архитектурой, аппаратно реконфигурируемой под выполняемый алгоритм [4–7].

В вычислителях с фиксированной архитектурой (многоядерные/многопотоковые процессоры и GPGPU) возможности повышения производительности ограничены особенностями конкретного вычислителя: числом вычислительных блоков; существующими связями между вычислительными блоками и их быстродействием; объёмом локальной/распределенной памяти, шириной каналов доступа к памяти; быстродействием и производительностью памяти. Эти ограничения являются фиксированными для конкретного вычислителя с фиксированной архитектурой и определяют зависимость между производительностью вычислителя и типом решаемой на нем задачи.

Вычислители с реконфигурируемой внутренней архитектурой, подстраиваемой под задачу, традиционно реализуемые на FPGA, во многом лишены указанных выше недостатков фиксированных архитектур. Однако одним из недостатков использования ускорителей вычислений на FPGA как для матричных операций, так и для других задач, является сложность традиционной процедуры разработки, основанной как на схемном описании, так и на использовании языков описания аппаратуры, например, языков VHDL, Verilog HDL, System Verilog [8–11].

Процесс создания вычислителя, аппаратно оптимизированного под алгоритм решаемой задачи, трудоёмок и требует существенных временных затрат как на этапе его разработки, так и на этапе его отладки [5]. Это зачастую не позволяет провести исследование и сравнительный анализ разных вариантов аппаратных реализаций алгоритма решаемой задачи. Что приводит к аппаратным решениям по производительности близким к решениям на основе вычислителей с фиксированными архитектурами [12, 13].

К настоящему времени существует ряд работ, связанных с созданием аппаратных реализаций вычислительно сложных алгоритмов и демонстрирующих указанные выше особенности и недостатки.

Так, в [3] описана попытка использования FPGA для выполнения операций с матрицами и векторами и сравнение эффективности с реализациями на базе Цифровых Сигнальных Процессоров (DSP), GPGPU и Специализированных Интегральных Схем (ASIC). Сделан вывод о применимости FPGA для решения подобных вычислительных задач, а основным недостатком использования FPGA названа сложность разработки устройства с использованием языков описания аппаратуры HDL. При этом в работе, в силу заявленной сложности создания вариантов аппаратных реализаций, не проведены исследования и оптимизация аппаратных реализаций.

В [4] сравнивается производительность FPGA и GPGPU при решении задачи умножения разреженной матрицы на вектор. Для GPGPU используется решение, основанное на технологии CUDA. В работе приведена сравнительная таблица производительности решения поставленной задачи на GPGPU и на базе FPGA, показывающая, что предложенная аппаратная реализация алгоритма умножения разреженной матрицы на вектор несколько производительнее реализации на GPGPU.

В [5] представлено сравнение производительности GPGPU, многоядерных систем и аппаратной реализации на FPGA при решении задачи перемножения матриц. Показано, что максимальная производительность достигается для решений на базе GPGPU и FPGA. При этом FPGA показали лучший результат по критерию энергоэффективности.

В [6] рассмотрены две реализации одного алгоритма умножения матриц на FPGA, созданные с использованием традиционного подхода к проектированию, включающему использование языков описания аппаратуры и схемного ввода. В статье показано, что полученные аппаратные реализации по производительности сравнимы с реализациями аналогичных алгоритмов на многоядерных системах.

В работе [7] предложена конвейерная архитектура матричного умножения на FPGA, созданная на базе схемного ввода. Особое внимание уделено оценке производительности передачи данных между подмодулями устройства. Оценка производительности проводилась в сравнении с оценкой производительности на базе моделирования в пакете MATLAB. В результате показано, что созданная реализация на FPGA имеет схожую производительность.

Современным подходом при создании аппаратной реализации алгоритма решаемой задачи является использование возможностей средств высокоуровневого синтеза – синтеза аппаратных решений описаний, созданных на высокоуровневых языках программирования, обычно на языках Си и C++. Подобные средства предоставляют как ведущие производители FPGA, такие как компания Xilinx и Intel PSG, так и компании, занимающиеся созданием средств разработки электронных устройств, например, компания Mentor Graphics. Подобные средства позволяют не только создать некоторое, базовое, не оптимизированное аппаратное решение на базе существующего описания на Си (или C++), но и провести исследование и сравнительный анализ различных вариантов аппаратных решений, отличающихся степенью параллелизма, количеством ступеней конвейера, интерфейсом к памяти данных, структурной организацией памяти данных и другими параметрами, связанными с эффективностью (критерии: производительность, аппаратные затраты) создаваемых аппаратных решений [14–16].

Эффективность, производительность и аппаратные затраты конечного результата, аппаратной реализации алгоритма решаемой задачи, существенно зависят от выбранного алгоритма решения задачи и подобранных параметров процедуры высокоуровневого синтеза. Процедура получения оптимального конечного результата не формализована, эвристическая и требует проведения исследований с использованием имитационного моделирования и сравнительного анализа [17–19].

Объект, предмет, методы, цель и средства исследования

Объектом исследования, результаты которого приведены в данной статье, является способ повышения производительности матричных вычислений.

Предмет исследования — операция суммирования транспонированных двумерных матриц размером *N* столбцов и *M* строк:

$$\mathbf{C}_{N \times M} = \mathbf{A}_{M \times N}^{\mathrm{T}} + \mathbf{B}_{M \times N}^{\mathrm{T}},$$

где С — двумерная выходная матрица размером $N \times M$; А и В — двумерные входные матрицы размером $N \times M$.

Пользуясь известными свойствами операции транспонирования матриц, алгоритм предмета исследования может быть представлен в модифицированном виде:

$$\mathbf{C}_{N \times M} = \left(\mathbf{A}_{M \times N} + \mathbf{B}_{M \times N}\right)^{\mathrm{T}},$$

где С — двумерная выходная матрица размером $N \times M$; А и В — двумерные входные матрицы размером $N \times M$.

Описание на языке Си прямого алгоритма предмета исследования приведено на рис. 1, где A_in и B_in – двумерные входные матрицы, размером $N \times M$; C_out – двумерная выходная матрица размером $M \times N$.

Функция transponse транспонирует входную матрицу, функция sum_matrix вычисляет сумму двух матриц с помощью двух вложенных циклов поэлементно, а T_SUM представляет собой функцию для решения целевой задачи с помощью предыдущих двух. При этом в функции T_SUM используются два временных (промежуточных, буферных) массива A_t и B_t для хранения транспонированных копий входных матриц A_in и B_in.

Описание на языке Си модифицированного алгоритма предмета исследования приведено на рис. 2, где A и B — двумерные входные матрицы, размером $N \times M$; C — двумерная выходная матрица размером $M \times N$.

В модифицированном алгоритме транспонирование выполняется при записи результатов суммирования двух матриц путем изменения порядка индексов. Поэтому в данной реализации алгоритма нет необходимости в буферных массивах, используемых для хранения промежуточных результатов.

Методы исследования:

 имитационное моделирование программной и аппаратной реализаций матричной операции суммирования двух транспонированных двумерных матриц (исходный алгоритм предмета исследования) и её модифицированной формы: транспонирования суммы двух двумерных матриц;

• сравнительный анализ по критериям: производительность и аппаратные затраты (для аппаратных реализаций).

Методика исследования включает следующие основные этапы:

• оценку производительности программной реализации суммирования двух транспонированных двумерных матриц;

• оценку производительности аппаратной реализации суммирования двух транспонированных двумерных матриц;

• оценку производительности программной реализации модифицированного алгоритма – операции транспонирования суммы двух двумерных матриц;

• оценку производительности аппаратной реализации модифицированного алгоритма — операции транспонирования суммы двух двумерных матриц;

• сравнительный анализ оценок, полученных при проведении исследования.

Цель исследования — выбор оптимального, по критерию производительность с учетом аппаратных затрат для аппаратных реализаций, способа реализации матричной операции суммирования двух транспонированных двумерных матриц.

Мера измерения производительности — промежуток времени, измеряемый в наносекундах, через который может быть запущена новая задача суммирования двух транспонированных матриц.

Мерами измерения аппаратных затрат являются: число встроенных модулей памяти (BRAM), необходимых для аппаратной реализации алгоритма, измеряемое в штуках; сумма числа триггеров (FF) и логических элементов (LUT), необходимых для аппаратной реализации алгоритма, измеряемая в штуках.

Исследования проводились для следующих наборов числа элементов квадратных матриц: 256×256, 512×512, 1024×1024, 2048×2048, 3072×3072 элементов. Пояснения по выбору числа элементов для проведения исследования: число элементов строк/столбцов выбрано равным степени двойки т. к. это максимальное число элементов для соответствующей разрядности адреса стро-

```
void transpose(data in_m[N][M], data out_m[M][N])
{
Loop_T1: for (int i = 0; i < N; i++)
Loop_T2 : for (int j = 0; j < M; j++)
out_m[j][i] = in_m[i][j];
}
void sum_matrix(data A_m[M][N], data B_m[M][N], data C_m[M][N])
{
Loop_Sum1: for(int j=0; j < N; j++)
Loop_Sum2: for( int i=0; i < M; i++)
C_m[i][j]=A_m[i][j] + B_m[i][j];
}
void T_SUM(data A_in[N][M], data B_in[N][M], data C_out[M][N]){
data A_t[M][N], B_t[M][N];
transpose(A_in, A_t);
transpose(B_in, B_t);
sum_matrix(A_t, B_t, C_out);
}</pre>
```





Puc. 2. Модифицированная форма алгоритма суммирования транспонированных двумерных матриц Fig. 2. Modified form of the algorithm for summing transposed two-dimensional matrices

ки/столбца; полученные результаты практически не зависят от задания числа элементов строк/ столбцов не равными степени двойки, например, такими как 255×255, 511×511 и далее, соответственно. Тип элементов матриц – целые, знаковые числа, разрядность 32 бита.

Средством для имитационного моделирования при программной реализации был выбран персональный компьютер (ПК) со следующими характеристиками: процессор – AMD Ryzen7 6800HS; оперативная память – 64 Гбайт, DDR5. Оболочка разработки (компилятор) для создания исполняемых файлов имитационных программных моделей – Microsoft Visual Studio 2022 Community.

Средством для аппаратной реализации, определяющим ограничения на доступные аппаратные ресурсы и задержки выполнения аппаратных функций, была выбрана микросхема (FPGA) компании Xilinx: xcku115-flva1517-2-е.

Средством для синтеза аппаратных реализаций на основе описания алгоритма на языке Си являлся пакет Vitis HLS 2022.2. Пакет позволяет осуществлять синтез устройства, получать оценки времени решения задачи, тактовой частоты устройства, а также выполнять имитационное моделирование для получения оценок производительности.

Процедура и результаты исследования

Для проведения исследования аппаратной и программной реализаций двух вариантов алгоритма суммирования двух транспонированных двумерных матриц на языке Си созданы описания, алгоритмические основы которых приведены на рис. 1, 2.

Для проведения исследования программной реализации создана имитационная модель, осуществляющая для каждого набора числа элементов (256×256, 512×512, 1024×1024, 2048×2048, 3072×3072) квадратных матриц:

— формирование двумерных квадратных матриц A и B, и их заполнение случайными данными;

 – многократный (*K* раз) запуск функций SUM_T и T_SUM, алгоритмы которых приведены на рис. 1, 2 соответственно;

 оценку времени выполнения функций SUM_T и T_SUM при каждом запуске и запись оценок в соответствующие массивы;

 нахождение K/2+1 статистики (медианы) среди элементов массивов с результатами оценок времени выполнения функций SUM_T и T_SUM, а также минимальной и максимальной оценок времени;

 контроль алгоритмической правильности работы функций SUM_T и T_SUM, осуществляемый путем сравнения результатов, полученных этими функциями;

 отображение минимальной, максимальной и медианной оценок времени выполнения функций SUM_T и T_SUM для текущего набора числа элементов в матрице.

Для аппаратной реализаций каждого из двух вариантов алгоритма суммирования двух транспонированных двумерных матриц в рамках пакета Vitis HLS 2022.2: созданы наборы решений, отличающиеся степенью параллелизма, коэффициентом конвейеризации и структурной организацией памяти хранения данных; проведены исследования и выбран период тактового сигнала, осуществляющего синхронизацию элементов аппаратной реализации; получены оценки производительности и аппаратных затрат созданных наборов аппаратных решений.

Проведен сравнительный анализ результатов исследований и выбран оптимальный, по критериям производительность/аппаратные затраты, алгоритм реализации суммирования двух транспонированных двумерных матриц и способ его реализации.

При создании набора аппаратных решений использованы следующие директивы пакета Vitis HLS:

– решения sol_16_6 (sol_16_8): степень параллелизма внутреннего цикла – 16, коэффициент конвейеризации внутреннего цикла – 3, входные массивы данных структурно преобразованы в 16 массивов по второму индексу, выходные массивы данных структурно преобразованы в 16 массивов по первому индексу. Период тактового сигнала задан равным 6 нс (8 нс);

– решения sol_32_6 (sol_32_8): степень параллелизма внутреннего цикла – 32, коэффициент конвейеризации внутреннего цикла – 3, входные массивы данных структурно преобразованы в 32 массива по второму индексу, выходные массивы данных структурно преобразованы в 32 массива по первому индексу. Период тактового сигнала задан равным 6 нс (8 нс).

Результаты исследования программной и аппаратной реализаций прямого алгоритма суммирования двух транспонированных двумерных матриц, представленные в виде графиков, приведены на рис. 3.

По оси абсцисс отложено число столбцов и строк матриц, использованных в исследовании. По оси ординат, представленной в логарифмическом масштабе, отложено время выполнения алгоритма в наносекундах.

На рисунке показаны графики зависимости времени выполнения алгоритма от размера матрицы для PC – программной реализации; base (8 ns) – аппаратной реализации без оптимизации, период тактового сигнала 8 нс; sol_16_6 (sol_16_8) и sol_32_6 (sol_32_8) – оптимизированных аппаратных решений, описанных выше.

Из анализа данных, представленных на рис. 3, следует:

- графики для решений sol_16_6 и sol_16_8 практически совпали;
- графики для решений sol_32_6 и sol_32_8 практически совпали;
- решение base является самым медленным;

– решение PC показывает большую производительность относительно решений sol_16_6 (sol_16_8) для матриц с числом строк = столбцов <= 512, и меньшую производительность относительно решений sol_32_6 (sol_32_8) для любого числа строк/столбцов матриц.

Оценка аппаратных затрат для указанных выше аппаратных решений приведено на рис. 4.





Рис. 3. Производительность для прямой формы алгоритма суммирования транспонированных двумерных матриц

Fig. 3. Performance for the direct form of the algorithm for summing transposed two-dimensional matrices



Рис. 4. Аппаратные затраты для прямой формы алгоритма суммирования транспонированных двумерных матриц

Fig. 4. Hardware costs for the direct form of the algorithm for summing transposed two-dimensional matrices

На оси ординат показана сумма использованных логических элементов и триггеров для соответствующей аппаратной реализации.

Из анализа рисунка можно заметить то, что потенциально более производительные варианты (у которых период тактовой частоты задан равным 6 нс) требуют больших аппаратных затрат. Однако из анализа рис. 3 следует, что оценки производительности, полученные после имитационного моделирования, зависят только от степени параллелизма и практически не зависят от установленного требования к периоду тактовой частоты.

Оценка числа внутренних блоков памяти, использованных для буферных массивов (см. рис. 1), показывает его квадратичную зависимость от числа строк/столбцов матриц. Поэтому, приведенные на рис. 3, 4 решения могут быть реализованы на микросхеме xcku115-flva1517-2-е, выбранной для имитационного моделирования, только для матриц с количеством строк = = столбцов <= 1024. Что ограничивает применимость прямого алгоритма суммирования двух транспонированных двумерных матриц для аппаратной реализации. Лучшей аппаратной реализацией прямой формы алгоритма, по критериям производительность/аппаратные затраты, является решение sol_32_8.





Fig. 5. Performance for a modified form of the algorithm for summing transposed two-dimensional matrices

Результаты исследования программной и аппаратной реализаций модифицированного алгоритма суммирования двух транспонированных двумерных матриц, представленные в виде графиков, приведены на рис. 5.

По оси абсцисс отложено число столбцов и строк матриц, использованных в исследовании. По оси ординат, представленной в логарифмическом масштабе, отложено время выполнения алгоритма в наносекундах.

Из анализа полученных результатов следует, что программное решение (PC) показывает меньшую производительность относительно всех аппаратных решений для любого числа строк/столбцов матриц; решения sol_32_6 (sol_32_8) показывают большую производительность в сравнении с решениями sol_16_6 (sol_16_8).

Оценки затрат аппаратных ресурсов, требуемых для аппаратной реализации решений модифицированного алгоритма, показывают аналогичный рис. 4 характер зависимости от числа строк/столбцов обрабатываемых матриц и заданных требований к периоду тактовой частоты. Абсолютные значения количества использованных логических элементов и триггеров для соответствующей аппаратной реализации модифицированного алгоритма примерно в два раза меньше, чем для аппаратной реализации прямого алгоритма (см. рис. 4). При этом модифицированный алгоритм не требует буферных матриц. Лучшей аппаратной реализацией модифицированной формы алгоритма, по критериям производительность/аппаратные затраты, является решение sol_32_8.

Обобщение полученных в результате исследования результатов приведено на рис. 6.

По оси абсцисс отложено число столбцов и строк матриц, использованных в исследовании. По оси ординат, представленной в логарифмическом масштабе, отложено время выполнения алгоритма в наносекундах.

На рисунке показаны графики зависимости времени выполнения алгоритма от размера матрицы для T_SUM_PC – программной реализации прямой формы алгоритма; SUM_T_PC – программной реализации модифицированной формы алгоритма; T_SUM_32_8 – аппаратной реализации прямой формы алгоритма (базируется на решении sol_32_8, приведенном на рис. 3); SUM_T_32_8 – аппаратной реализации модифицированной формы алгоритма (базируется на решении sol_32_8, приведенном на рис. 5).

Анализ полученных обобщенных результатов показывает:

• Модифицированный алгоритм суммирования двух транспонированных двумерных матриц позволяет как для программной, так и для аппаратной реализаций достичь примерно в два раза большей производительности относительно производительности прямого алгоритма для соответствующих решений.





Рис. 6. Обобщенные результаты исследования Fig. 6. Generalized results of the study

• Максимальную производительность дает аппаратное решение, основанное на модифицированной форме алгоритма — решение SUM_T_32_8. При этом выигрыш производительности аппаратного SUM_T_32_8 решения относительно лучшего программного решения увеличивается с увеличением числа строк/столбцов матриц, а сложность, число использованных логических элементов и триггеров аппаратного решения практически не изменяется.

Заключение

Цель проведения исследования достигнута — определен способ реализации операции суммирования двух транспонированных двумерных матриц, обеспечивающий максимальную производительность для заданных ограничений, и создано соответствующее аппаратное решение SUM_T_32_8.

Показано, что алгоритм реализации операции суммирования двух транспонированных двумерных матриц имеет существенное влияние как на производительность программного и аппаратных решений, так и на аппаратные затраты при аппаратной реализации.

В качестве дальнейшей работы в данном направлении целесообразно провести аналогичные исследования других часто используемых матричных операций.

СПИСОК ЛИТЕРАТУРЫ

1. Antonov A., Zaborovskiy V., Kiselev I. The reconfigurable computational modules in network-centric supercomputer systems // High Availability Systems. 2018. vol. 14. No. 3. Pp. 57–62. DOI: 10.18127/j20729472-201803-09

2. Kobayashi R., Oobata Y., Fujita N., Yamaguchi Y., Boku T. OpenCL-ready high speed FPGA network for reconfigurable high performance computing // Proc. of the Internat. Conf. on High Performance Computing in Asia-Pacific Region. 2018. Pp. 192–201. DOI: 10.1145/3149457.3149479

3. **Qasim S.M., Telba A.A., Al Mazroo A.** FPGA design and implementation of matrix multiplier architectures for image and signal processing applications // Internat. J. of Computer Science and Network Security. 2010. No. 10. Pp. 168–176.

4. Zhang Y., Shalabi Y.H., Jain R., Nagar K.K., Bakos J.D. FPGA vs. GPU for sparse matrix vector multiply // Internat. Conf. on Field-Programmable Technology. 2009. Pp. 255–262. DOI: 10.1109/FPT.2009.5377620

5. Tan Y., Imamura T., Mukunoki D. Design of an FPGA-based matrix multiplier with task parallelism // Parallel Computing: Technology Trends. 2020. vol. 36. Pp. 241–250. DOI: 10.3233/APC200047

6. Kumar V., Joshi S., Patkar S., Narayanan H. FPGA based high performance double-precision matrix multiplication // Internat. J. of Parallel Programming. 2010. No. 38. Pp. 322–338. DOI: 10.1007/s10766-010-0131-8

7. Jiang J., Mirian V., Tang K., Chow P., Xing Z. Matrix multiplication based on scalable macro-pipelined FPGA accelerator architecture // Internat. Conf. on Reconfigurable Computing and FPGAs. 2009. Pp. 48–53. DOI: 10.1109/ReConFig.2009.30

8. Abbaszadeh A., Iakymchuk T., Bataller-Mompeán M., Frances-Villora J.V., Rosado A. Anscalable matrix computing unit architecture for FPGA and SCUMO user design interface // Electronics. 2019. vol. 8. P. 94. DOI: 10.3390/electronics8010094

9. Antonov A., Zaborovskij V., Kisilev I. Developing a new generation of reconfigurable heterogeneous distributed high performance computing system // Proc. of Internat. Scientific Conf. on Telecommunications, Computing and Control. Smart Innovation, Systems and Technologies. Springer, 2021. vol 220. DOI: 10.1007/978-981-33-6632-9_22

10. Antonov A., Besedin D., Filippov A. Research of the efficiency of high-level synthesis tool for FPGA based hardware implementation of some basic algorithms for the big data analysis and management tasks // 2020 26th Conf. of Open Innovations Association (FRUCT). Yaroslavl, Russia, 2020. Pp. 1–7. DOI: 10.23919/ FRUCT48808.2020.9087355

11. Kalyaev I., Antonov A., Zaborovskij V. Architecture of reconfigurable heterogeneous distributed supercomputer system for solving problems of intelligent data processing in the era of digital transformation of the economy // Cybersecurity Issues. 2019. Pp. 2–11. DOI: 10.21681/2311-3456-2019-5-02-11

12. Asch M., Moore T., Badia R., et al. Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry // The Internat. J. of High Performance Computing Applications. 2018. vol. 4. No. 32. Pp. 435–479. DOI: 10.1177/1094342018778123

13. **Haidar A.** Investigating power capping toward energy-efficient scientific applications // Concurrency and Computation Practice and Experience. 2018. Pp. 1–14. DOI: 10.1002/cpe.4485

14. Antonov A., Zaborovsky V., Polyanskiy V. Neural computations in control problems: Aspects of computability and spatial-time characterization of cognitive functions // J. of Physics: Conference Series. 2021. vol. 1864. No. 1. DOI: 10.1088/1742-6596/1864/1/012104

15. **Dongarra J.** Race to exascale // Computing in Science and Engineering. 2019. vol. 21. No. 1. Pp. 4–5. DOI: 10.1109/MCSE.2018.2882574

16. Harris A. Exascale models of stellar explosions: Quintessential multi-physics simulation // The Internat. J. of High Performance Computing Applications. 2021. DOI: 10.1177/10943420211027937

17. Le Fèvre V. Comparing the performance of rigid, moldable and grid-shaped applications on failureprone HPC platforms // Parallel Computing. 2019. vol. 85. Pp. 1–12. DOI: 10.1016/j.parco.2019.02.002

18. Usman A., Fathy A., Aiiad A., Abdullah A. Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems // IEEE Access. 2018. No. 6. Pp. 23095–23107. DOI: 10.1109/ACCESS.2018.2823299

19. Mantovani F., Calore E. Performance and power analysis of HPC workloads on heterogeneous multi-node clusters // Low Power Electron. 2018. vol. 2. No. 8. Pp. 1–14. DOI:10.3390/jlpea8020013

REFERENCES

1. Antonov A., Zaborovskiy V., Kiselev I. The reconfigurable computational modules in network-centric supercomputer systems. *High Availability Systems*, 2018, vol. 14, No. 3, Pp. 57–62. (rus). DOI: 10.18127/j20729472-201803-09

2. Kobayashi R., Oobata Y., Fujita N., Yamaguchi Y., Boku T. OpenCL-ready high speed FPGA network for reconfigurable high performance computing. *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, 2018, Pp. 192–201. DOI: 10.1145/3149457.3149479

3. **Qasim S.M., Telba A.A., Al Mazroo A.** FPGA design and implementation of matrix multiplier architectures for image and signal processing applications. *International Journal of Computer Science and Network Security*, 2010, No. 10, Pp. 168–176.

4. Zhang Y., Shalabi Y.H., Jain R., Nagar K.K., Bakos J.D. FPGA vs. GPU for sparse matrix vector multiply. *International Conference on Field-Programmable Technology*, 2009, Pp. 255–262. DOI: 10.1109/FPT.2009.5377620

5. Tan Y., Imamura T., Mukunoki D. Design of an FPGA-based matrix multiplier with task parallelism. *Parallel Computing: Technology Trends*, 2020, vol. 36, Pp. 241–250. DOI: 10.3233/APC200047

6. Kumar V., Joshi S., Patkar S., Narayanan H. FPGA based high performance double-precision matrix multiplication. *International Journal of Parallel Programming*, 2010, No. 38, Pp. 322–338. DOI: 10.1007/s10766-010-0131-8

7. Jiang J., Mirian V., Tang K., Chow P., Xing Z. Matrix multiplication based on scalable macro-pipelined FPGA accelerator architecture. *International Conference on Reconfigurable Computing and FPGAs*, 2009, Pp. 48–53. DOI: 10.1109/ReConFig.2009.30

8. Abbaszadeh A., Iakymchuk T., Bataller-Mompeán M., Frances-Villora J.V., Rosado A. Anscalable matrix computing unit architecture for FPGA and SCUMO user design interface. *Electronics*, 2019, vol. 8. Pp. 94. DOI: 10.3390/electronics8010094

9. Antonov A., Zaborovskij V., Kisilev I. Developing a new generation of reconfigurable heterogeneous distributed high performance computing system. *Proceedings of International Scientific Conference on Telecommunications, Computing and Control. Smart Innovation, Systems and Technologies.* Springer, 2021, vol. 220. DOI: 10.1007/978-981-33-6632-9_22

10. Antonov A., Besedin D., Filippov A. Research of the efficiency of high-level synthesis tool for FPGA based hardware implementation of some basic algorithms for the big data analysis and management tasks. 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 2020, Pp. 1–7. DOI: 10.23919/FRUCT48808.2020.9087355

11. **Kalyaev I., Antonov A., Zaborovskij V.** Architecture of reconfigurable heterogeneous distributed supercomputer system for solving problems of intelligent data processing in the era of digital transformation of the economy. *Cybersecurity Issues*, 2019, Pp. 2–11. DOI: 10.21681/2311-3456-2019-5-02-11

12. Asch M., Moore T., Badia R., et al. Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry. *The International Journal of High Performance Computing Applications*, 2018, vol. 4, No. 32, Pp. 435–479. DOI: 10.1177/1094342018778123

13. Haidar A. Investigating power capping toward energy-efficient scientific applications. *Concurrency* and Computation Practice and Experience, 2018, Pp. 1–14. DOI: 10.1002/cpe.4485

14. Antonov A., Zaborovsky V., Polyanskiy V. Neural computations in control problems: Aspects of computability and spatial-time characterization of cognitive functions. *Journal of Physics: Conference Series*, 2021, vol. 1864, No. 1. DOI: 10.1088/1742-6596/1864/1/012104

15. **Dongarra J.** Race to exascale. *Computing in Science and Engineering*, 2019, vol. 21, No. 1, Pp. 4–5. DOI: 10.1109/MCSE.2018.2882574

16. Harris A. Exascale models of stellar explosions: Quintessential multi-physics simulation. *The International Journal of High Performance Computing Applications*, 2021. DOI: 10.1177/10943420211027937

17. Le Fèvre V. Comparing the performance of rigid, moldable and grid-shaped applications on failure-prone HPC platforms. *Parallel Computing*, 2019, vol. 85, Pp. 1–12. DOI: 10.1016/j.parco.2019.02.002

18. Usman A., Fathy A., Aiiad A., Abdullah A. Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems. *IEEE Access*, 2018, No. 6, Pp. 23095–23107. DOI: 10.1109/ACCESS.2018.2823299

19. **Mantovani F., Calore E.** Performance and power analysis of HPC workloads on heterogeneous multi-node clusters. *Low Power Electron*, 2018, vol. 2, No. 8, Pp. 1–14. DOI:10.3390/jlpea8020013

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Антонов Александр Петрович Alexander P. Antonov E-mail: antonov@eda-lab.ftk.spbstu.ru

Беседин Денис Сергеевич Denis S. Besedin E-mail: nero1310@yandex.ru

Филиппов Алексей Семенович Alexey S. Filippov E-mail: alexey.s.filippov@gmail.com

Поступила: 03.12.2022; Одобрена: 26.12.2022; Принята: 12.01.2023. Submitted: 03.12.2022; Approved: 26.12.2022; Accepted: 12.01.2023. The Seventh Conference on Software Engineering and Information Management (SEIM-2022) Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

Research article DOI: https://doi.org/10.18721/JCSTCS.15405 UDC 004.852



TOKEN-WISE APPROACH TO SPAN-BASED QUESTION ANSWERING

A.A. Pismenny¹ \bowtie , E.A. Sokolov²

¹ Yandex, Moscow, Russian Federation; ² National Research University 'Higher School of Economics', Moscow, Russian Federation

□ pismennyy.aleks@gmail.com

Abstract. Language model pre-training has led to significant success in a wide range of natural language processing problems. It was shown that modern deep contextual language models need only a small number of new parameters for fine-tuning due to the power of the base model. Nevertheless, the statement of the problem itself makes it possible to search the new approaches. Our experiments relate to the span-based question answering, one of machine reading comprehension (MRC) tasks. Recent works use loss functions that require the model to predict start and end positions of the answer in a contextual document. We propose a new loss that additionally requires the model to correctly predict whether each token is contained in the answer. Our hypothesis is that explicit using of this information can help the model to learn more dependencies from data. Our solution also includes a new span's ranking and a no-answer examples selection scheme. We also propose approaches of accounting for information about relative positions of tokens in the dependency trees and the types of dependencies in relation to syntax-guided attention. The experiments showed that our approaches increase the quality of BERT-like models on SQuAD datasets.

Keywords: machine learning, natural language processing, question answering, machine reading comprehension, dependency parsing

Citation: Pismenny A.A., Sokolov E.A. Token-wise approach to span-based question answering. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 64–72. DOI: 10.18721/JCSTCS.15405

The Seventh Conference on Software Engineering and Information Management (SEIM-2022)

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15405 УДК 004.852



ПОТОКЕННЫЙ ПОДХОД К ПОИСКУ ДИАПАЗОНА ПРАВИЛЬНОГО ОТВЕТА В ВОПРОСНО-ОТВЕТНЫХ СИСТЕМАХ

А.А. Письменный¹ ⊠, Е.А. Соколов²

¹ Яндекс, Москва, Российская Федерация; ² Национальный исследовательский университет «Высшая школа экономики», Москва, Российская Федерация

^{III} pismennyy.aleks@gmail.com

Аннотация. Использование предобученных языковых моделей привело к значительному успеху в решении широкого круга задач обработки естественного языка. Показано, что современным глубоким языковым моделям достаточно лишь небольшого количества дополнительных параметров для дообучения, что достигается за счет мощности базовой модели. Тем не менее сама постановка задачи дообучения позволяет искать новые подходы. Наши эксперименты связаны с задачей поиска диапазона правильного ответа, одним из вариантов задачи машинного понимания прочитанного. Во многих современных работах для данной задачи используются функции потерь, которые предполагают, что модель предсказывает только позиции начала и конца правильного ответа в документе. В данной статье предложена новая функция потерь, направленная на то, чтобы модель правильно предсказывала, содержится ли каждый токен в правильном ответе. Наша гипотеза состоит в том, что явное использование этой информации может помочь модели извлечь больше зависимостей из данных. Предложенное решение также включает в себя новую схему ранжирования диапазонов и схему выбора примеров без правильного ответа. Предложены подходы к учету информации о взаимном расположении токенов в деревьях зависимостей и типах зависимостей вместе с использованием синтаксически управляемого механизма внимания. Эксперименты показывают, что предложенные подходы повышают результат для решений, основанных на модели BERT (Bidirectional Encoder Representations from Transformers), на наборах данных SQUAD (Stanford Question Answering Dataset).

Ключевые слова: машинное обучение, обработка естественного языка, вопросно-ответные системы, машинное понимание прочитанного, синтаксический разбор зависимостей

Для цитирования: Pismenny A.A., Sokolov E.A. Token-wise approach to span-based question answering // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 64–72. DOI: 10.18721/JCSTCS.15405

Introduction

Question Answering (QA) is an important natural language understanding problem. One of examples of related tasks is span-based QA, where an answer is looked up as a sub-sequence in a contextual document. Selecting the span of the answer to a question might be used as a final step in the open-domain QA pipeline, therefore, advances in span-based QA can be applied for real-world question-answering problems [1]. Also, this task is one of the options of Machine Reading Comprehension (MRC) problem, where the task is to read the text and answer questions based on it (in this case the answering form is span selection). In this paper, we focus on a span-based QA task. Given a question and a passage containing the information for understanding, the task is to predict the contiguous span of answer text in this passage. Additional difficulty may lie in identifying those examples in which the passage does not contain a right answer (no-answer examples). We propose new a training objective and suitable answer selection scheme. Our approach is applied to a pre-trained BERT model [2]. Evaluation on challenging question answering

tasks shows than the proposed solution benefits the model performance. Our additional experiments focus on using information from dependency parsing. Our contributions are as follows:

1. We propose a new approach for span-based QA that takes into account more supervised information from the dataset.

2. We propose new approaches of accounting for information about relative positions of tokens in the dependency trees and the types of dependencies.

3. We validate our approaches on both SQuAD datasets and show that they are significantly better than BERT baseline.

Related work

Contextual language model pre-training has significantly improved performance in a wide range of NLU tasks including question-answering benchmarks. Some important examples are Embedding from Language models (ELMo) [3], Bidirectional Encoder Representations from Transformers (BERT) [2] and BERT-based approaches, such as RoBERTa [4] and ALBERT [5]. Many of these models use the self-attention mechanism proposed in Vaswani et al. [6]. For span-based QA, this mechanism can provide bidirectional cross attention between the passage and the question. Many works have been devoted to the inclusion of syntactic/semantic information for solving NLP tasks. For example, Zhang et al. [7] experimented with adding explicit syntactic constraints to the attention mechanism by focusing on ancestors in the dependency-parsing tree. Zhang et al. [8] used an additional module that processes information received from semantic role labeling.

Proposed methods

Training objective and prediction. The standard training objective function for span-based question answering is the negative sum of the log probabilities of the predicted distributions indexed by true start and end indices averaged over all the training examples (Seo et al. [9], Yu et al. [10], Devlin et al. [2], Liu et al. [4], Lan et al. [5] and others). On the one hand, the information about tokens between the start and end answer tokens is also used in these models. On the other hand, it is possible to take this information into account more explicitly. Our hypothesis is that using token-wise labels can help the model to learn more dependencies from data. Our model predicts whether each particular token is contained in the answer. The important question in this case is how to select answer on test data. The proposed solution is described below.

Similarly to Devlin et al. [2], we use a start vector $S \in \mathbb{R}^d$ and an end vector $E \in \mathbb{R}^d$ (fully connected layers over the encoder output) during fine-tuning. The start and end logits for one example are computed by

$$s = SH^T$$
, $e = EH^T$,

where $H \in \mathbb{R}^{n \times d}$ is encoder output, *n* is input length, *d* is hidden size.

We use the additional vector $B \in \mathbb{R}^d$ for binary classification task. Suppose the correct answer is spanned from i^{th} to j^{th} token. Denote logits for the new vector by $b = BH^T$. The loss function is

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_e \mathcal{L}_e + \lambda_b \mathcal{L}_b,$$

where $\lambda_s, \lambda_e, \lambda_b$ are hyperparameters. The components of the loss function are defined as follows

$$\mathcal{L}_{s} = -\log(softmax(s)_{i}), \ \mathcal{L}_{e} = -\log(softmax(e)_{j}),$$



Fig. 1. DT-BERT model

$$\mathcal{L}_{b} = -\frac{1}{n} \left[\sum_{k \in \overline{i}, j} \log(\sigma(b_{k})) + \sum_{k \in \overline{0, n-1}, k \notin \overline{i}, j} \log(1 - \sigma(b_{k})) \right].$$

The DT-BERT (Double Task BERT) model is shown in Fig. 1. The score of a candidate span from position i to position j is defined as

$$T(i, j) = s_i + e_j + \frac{\alpha}{j - i + 1} \sum_{k \in \overline{i, j}} b_k$$

where α is a hyperparameter that is selected on the validation set. The maximum scoring span where $j \ge i$ is used as a prediction.

We also experimented with maximizing the following function:

$$P(i,j) = s_i + e_j + \frac{1}{n} \left[\sum_{k \in \overline{i}, j} \log(\sigma(b_k)) + \sum_{k \in \overline{0, n-1}, k \neq \overline{i}, j} \log(1 - \sigma(b_k)) \right],$$

which is equivalent to finding the span with the minimal value of the loss function (similarly to $s_i + e_j$ maximizing for the base model). In this case, the model performed worse. Replacing 1/n coefficient with a hyperparameter, which is selected on the validation set, gives comparable results.

The extra layer *B* can also be used to predict unanswerable questions. As usual for BERT-based models, we treat questions that do not have an answer as having an answer span with start and end at the [CLS] (null) token. We predict a non-null answer when $\max_{i \in \overline{1,n-1}} \sigma(b_i) > \sigma(b_0) + \tau$, where the threshold τ is selected on the validation set (instead of $\max_{i \leq j} \{s_i + e_j\} > s_0 + e_0 + \tau$ in [2]). For the task with unanswerable questions we first optimize the EM metric on the answerable questions by choosing α , and then we choose the optimal threshold τ on all questions for fixed α value (also for EM).

Using information from dependency parsing. We adopt the self-attention layer modification proposed in [7]. Given input token sequence $S = s_1, s_2, ..., s_n$, where *n* denotes the sequence length, they derive the ancestor node set P_i for each word s_i according to the dependency tree. Then a $n \times n$ mask *M* is used to add explicit syntactic constraints into attention mechanism (*X* is layer input, for example, the batch size is 1):

Attention
$$(Q, K, V) = \operatorname{softmax} \left(\frac{QK^T + M}{\sqrt{d_k}} \right) V$$
,
 $M_{i,j} = \begin{cases} 0, & \text{if } j \in P_i \text{ or } j = i, \\ -\infty, & \text{otherwise} \end{cases}$
 $Q = W^Q X, K = W^K X, V = W^V X, X \in \mathbb{R}^{n \times d}.$

We propose to supplement this mechanism with the information about the distance from each token to the root of the dependency tree corresponding to it (depth in a tree). We assume that this is sufficient information to understand the relative positions if we use described masking. The embeddings corresponding to each depth value are added to the input of dependency-based attention module:

$$Q = W^{\mathcal{Q}}(X+D), K = W^{\mathcal{K}}(X+D), V = W^{\mathcal{V}}(X+D),$$

where $D \in \mathbb{R}^{n \times d}$. If the level of the *i*th token is *k*, then D[i] = E[k], where $E \in \mathbb{R}^{m \times d}$, *m* is the maximum depth value, *E* is the learnable embeddings matrix for all depths, E[k] is embedding for *k*th level.

We are experimenting with two approaches: in the first case X is BERT encoder output (similarly to [7]), in the second case X is a matrix of relations in dependency trees. Specifically, the i^{th} row of the matrix is the embedding of the dependency between the i^{th} token and its parent in the dependency tree¹.

We use special dependency embeddings for special tokens [CLS], [SEP], and [PAD]. Each of such tokens corresponds to a separate dependency tree. For proposed approach we assume that each row of Attention() output is some embedding of the path from the "root" of the sentence to the corresponding token. The depth embeddings and the dependency embeddings are learned together with the model. In both cases, the additional module repeats the Transformer encoder block's architecture. For the first model dependency-based module's output is averaged with the BERT encoder output with weights $w_1 = w_2 = 0.5$, for the second we use concatenation and slightly increase the dimension of the linear layers *S*, *E* and *B*.

Experiments

Dataset and evaluation. Our experiments are carried on SQuAD 1.1 and SQuAD 2.0 datasets [11, 12] for span-based question answering. The SQuAD 2.0 task extends the SQuAD 1.1 by allowing for the possibility that no answer exists in the provided passage. The dataset statistics are described in Table 1 and Table 2. The Dev set is another name for validation set.

Table 1

Dataset statistics of SQuAD 1.1

	Train	Dev	Test
Num of examples	87.599	10.570	9.533

¹ Most of the dependencies are covered here: https://universaldependencies.org/u/dep/all.html

	Train	Dev	Test
Total examples	130.319	11.873	8.862
Examples with answer	86.821	5.928	4.530
Examples w/o answer	43.498	5.945	4.332

Dataset statistics of SQuAD 2.0

Two official metrics are used to evaluate the model performance on SQuAD datasets: Exact Match (EM) and a softer metric of F1 score (they were used in original papers [11, 12] and also in the official leaderboard). The EM measures the percentage of predictions that match any one of the ground truth answers exactly. The F1 score measures the overlap between the prediction and ground truth answer (for one example). The prediction and the ground truth are presented as bags of tokens, and their F1 is computed. The average F1 is calculated for all examples; in each case, the maximum F1 for all ground truth answers is taken. For unanswerable examples, EM and F1 are equal.

Implementation. We adopted the BERT-base-cased and BERT-large-cased-whole-word-masking² (BERT wwm) as the baselines. We used one training epoch for BERT-wwm model on SQuAD 1.1. For BERT-base-cased training, we used 3 epochs on SQuAD 1.1 and 4 epochs on SQuAD 2.0. The number of epochs was chosen from {1, 2, 3, 4} using learning rate $5 \cdot 10^{(-5)}$ to maximize the performance of our baselines. The initial learning rate was selected from $\{3 \cdot 10^{-5}, 4 \cdot 10^{-5}, 5 \cdot 10^{-5}, 6 \cdot 10^{-5}, 7.5 \cdot 10^{-5}\}$. We used the linear warmup for the first 10 % of steps followed by a linear decay to 0; L2 weight decay coefficient was 0.01. We set $\lambda_s = \lambda_e = \lambda_b = 1/3$ for proposed loss and $\lambda_s = \lambda_e = 1/2$, $\lambda_b = 0$ for baseline loss; our experiments with BERT-base on SQuAD 1.1 were done with double (× 2) weights λ_s , λ_e , λ_b . We used the WordPiece embeddings [13]; the maximum input length was set to 384. Our implementation was based on the Pytorch implementation of BERT³ and Spacy dependency parser⁴ (from "en_core_web_lg" model). The hidden size in dependency-based module for syntactic path encoding was set to 30, the number of attention heads was 3, and the intermediate size was 90.

Results. Table 3 shows SQuAD 1.1 Dev set results. Limited testing on this dataset is due to the fact that getting results on a test dataset is not currently available. The DT-BERT model outperforms the baselines in the task of choosing the correct span.

Table 3

System	EM	F1
BERT-Base [2]	80.8	88.5
BERT-Base (our baseline)	80.94	88.67
DT on BERT-Base	82.04	88.81
BERT wwm (google-research-github ⁵)	86.7	92.9
BERT wwm (our baseline)	87.38	93.36
DT on BERT wwm	87.87	93.54

SQuAD 1.1 Dev results

DT-BERT results on SQuAD 2.0 compared to prior leaderboard "BERT-base" entries and published BERT-base results are shown in Table 4.

Table 2

² https://github.com/google-research/bert/blob/master/README.md

³ https://github.com/huggingface/transformers

⁴ https://spacy.io/models/en

⁵ https://github.com/google-research/bert/blob/master/README.md

Table 4

System	D	ev	Test		
System	EM	F1	EM	F1	
BERT-Base [14]	73.66	76.30	_	_	
BERT-Base (best on leaderboard)	_	_	73.10	76.24	
BERT-Base (our baseline)	73.74	76.71	73.30	76.28	
DT on BERT-Base	74.48	77.54	74.77	77.71	

SQuAD 2.0 results. The results of BERT on the Dev set and Test set taken from Yang et al. [14] and SQuAD 2.0 leaderboard⁶ respectively

We find the proposed way to encode the relative position in dependency trees and the proposed encoding of the syntactic path useful as well.

Table 5

Results of models using dependency parsing, SQuAD 2.0

System	Dev		Test		
System	EM	F1	EM	F1	
DT-BERT	74.48	77.54	74.77	77.71	
+ Mask layer, input:					
relation embs + depth embs	74.96	77.64	_	_	
BERT output	75.20	78.25	_	_	
BERT output + depth embs	75.35	78.24	75.47	78.23	

We also evaluated the separate contribution of the proposed span's scoring, no-answer detection and mask layer over syntactic embeddings.

Table 6

SQuAD 2.0 Dev, ablation results. In the last version (last row), additional embeddings are not processed by the mask layer

System	EM	F1
DT-BERT	74.48	77.54
DT-BERT, w/o new no-answer detection	74.02	76.93
DT-BERT, w/o new span scoring	74.08	77.44
DT-BERT, w/o both (only new loss)	73.62	76.81
DT-BERT + Mask layer, input: relation embs + depth embs	74.96	77.64
DT-BERT + Mask layer, input: relation embs + depth embs, w/o mask layer	74.70	77.28

We used the McNemar's test [15] to test the statistical significance of SQuAD 1.1 and SQuAD 2.0 Dev set results (SQuAD test data are not publicly available). This test can be adapted for classification tasks [16], and span-based QA [17]. Our models are better than the baselines with p-value < 0.05.

⁶ https://rajpurkar.github.io/SQuAD-explorer/

Conclusion

We proposed a new approach for span-based question answering task, including new loss function, span's ranking and no-answer examples selection. The experiments on two widely used benchmarks showed improving performance both for finding correct answers and for identifying unanswerable questions. The proposed solution requires a minimum number of extra parameters and can also be used for any model that encodes input sequence tokens. The solution can be further improved by using more powerful encoders or syntactic information as demonstrated. The proposed approaches to its accounting also allow improving the results.

REFERENCES

1. Yang W., Xie Y., Lin A., Li X., Tan L., Xiong K., Li M., Lin J. End-to-end open-domain question answering with bertserini. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, Pp. 72–77.

2. Devlin J., Chang M.-W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1 (Long and Short Papers), 2019, Pp. 4171–4186.

3. Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. Deep contextualized word representations. *Proceedings of NAACLHLT*, 2018, Pp. 2227–2237.

4. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692 (2019)*.

5. Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations*, 2019.

6. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, Pp. 5998–6008.

7. Zhang Z., Wu Y., Zhou J., Duan S., Zhao H., Wang R. Sg-net: Syntax-guided machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, Pp. 9636–9643.

8. Zhang Z., Wu Y., Zhao H., Li Z., Zhang S., Zhou X., Zhou X. Semantics-aware bert for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, Pp. 9628–9635.

9. Seo M., Kembhavi A., Farhadi A., Hajishirzi H. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603 (2016)*.

10. Yu A.W., Dohan D., Luong M.-T., Zhao R., Chen K., Norouzi M., Le Q.V. Qanet: Combining local convolution with global self-attention for reading comprehension. *International Conference on Learning Representations*, 2018.

11. **Rajpurkar P., Zhang J., Lopyrev K., Liang P.** Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, Pp. 2383–2392.

12. **Rajpurkar P., Jia R., Liang P.** Know what you don't know: Unanswerable questions for squad. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (vol. 2: Short Papers), 2018, Pp. 784–789.

13. Wu Y., Schuster M., Chen Z., Le Q.V., Norouzi M., Macherey W., Krikun M., Cao Y., Gao Q., Macherey K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

14. Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R.R., Le Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 2019, Pp. 5753–5763. Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

15. **McNemar Q.** Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12 (1947): 153–157.

16. **Dietterich T.G.** Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10 (1998): 1895–1923.

17. Zhang Z., Yang J., Zhao H. Retrospective reader for machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, Pp. 14506–14514.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Письменный Алексей Алексеевич Alexey A. Pismenny E-mail: pismennyy.aleks@gmail.com

Соколов Евгений Андреевич Evgeny A. Sokolov E-mail: sokolov.evg@gmail.com

Поступила: 30.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023. Submitted: 30.11.2022; Approved: 26.12.2022; Accepted: 12.01.2022.
Research article DOI: https://doi.org/10.18721/JCSTCS.15406 UDC 004.8



MULTI-CHANNEL TRANSFORMER: A TRANSFORMER-BASED MODEL FOR MULTI-SPEAKER SPEECH RECOGNITION

*E.S. Fadeeva*¹ , *V.A. Ershov*² ^{1,2} Yandex LLC, St. Petersburg, Russian Federation noxoomo@yandex-team.ru

Abstract. Most of the modern approaches to multi-speaker speech recognition are either not applicable in case of overlapping speech or require a lot of time to run, which can be critical, for example, in case of real-time speech recognition. In this paper, a transformer-based end-to-end model for overlapping speech recognition is presented. It is implemented by using a generalization of the standard approach to speech recognition. The introduced model achieves results comparable in quality to modern state-of-the-art models, but requires less model calls, which speeds up the inference. In addition, a procedure for generating synthetic data for model training is described. This procedure allows to compensate for the lack of real multi-speaker speech training data by creating a stream of data from the initial collection.

Keywords: speech recognition, multi-speaker speech recognition, diarization, speech separation, voice technologies

Citation: Fadeeva E.S., Ershov V.A. Multi-channel transformer: A transformer-based model for multi-speaker speech recognition. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 73–85. DOI: 10.18721/JCSTCS.15406

Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15406 УДК 004.8



МНОГОКАНАЛЬНЫЙ ТРАНСФОРМЕР: МОДЕЛЬ ДЛЯ РАСПОЗНАВАНИЯ МНОГОГОЛОСНОЙ РЕЧИ, ОСНОВАННАЯ НА АРХИТЕКТУРЕ ТРАНСФОРМЕР

Е.С. Фадеева¹, В.А. Ершов² ⊠

^{1,2} ООО «Яндекс», Санкт-Петербург, Российская Федерация похоото@yandex-team.ru

Аннотация. Многие современные подходы для решения задачи распознавания многоголосной речи либо не предназначены для работы с пересекающейся речью, либо требуют много времени для запуска, что может быть критичным, например, в случае распознавания речи в реальном времени. В статье предложена трансформерная endto-end модель для распознавания многоголосной речи с возможными пересечениями. Предложенная архитектура является обобщением архитектуры из стандартного подхода к распознаванию речи. Такая модель позволяет достичь результатов, сопоставимых по качеству с современными решениями, но требует меньше запусков модели для получения текстового распознавания многоголосной речи, что ускоряет время работы такой системы. Описана процедура генерации синтетических данных для обучения модели. Эта процедура позволяет компенсировать отсутствие реальных данных для обучения модели для распознавания многоголосной речи путем создания потока данных из первоначального набора.

Ключевые слова: распознавание речи, распознавание многоголосной речи, диаризация, разделение речи, голосовые технологии

Для цитирования: Fadeeva E.S., Ershov V.A. Multi-channel transformer: A transformer-based model for multi-speaker speech recognition // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 73–85. DOI: 10.18721/JCSTCS.15406

Introduction

Speech recognition is a problem of determining a text spoken on an audio signal. Voice technologies are used in many aspects of human life: voice assistants, smartphones for blind people, transcribing voice messages.

Periodically, a situation of multi-speaker speech arises in speech technologies, that is, when several people are speaking on an audio recording. Solutions to the problem of voice recognition in such situations usually give out the whole speech in one text, without identifying the specific speakers of the recognized words. But if it is necessary to determine the spoken texts for each speaker separately, the problem of multi-speaker speech recognition arises.

Systems for automatic multi-speaker speech recognition can be used for mark-up and further analysis of various audio data, for example, dialogues at conferences or personal meetings.

In the general case of the problem of multi-speaker speech recognition, the number of people participating in the conversation is not known in advance, as well as any additional information about the speakers is unknown. Another difficulty of the problem is the periodically occurring situation of overlapping speech (when people interrupt each other).

Thus, in the most general formulation of the problem of multi-speaker speech recognition, an audio signal comes to the input of the system, where the speech of several people sounds (the number of which

is unknown), and the system should output several voice-recognition texts of each of the participants in the conversation.

Diarization is a problem of dividing input audio recording into several segments with a designated unique number for each the speaker whose voice sounds throughout this segment. In situations of overlapping speech, these segments should overlap.

Multi-speaker speech recognition is closely related to the problem of diarization, as is its combination with a problem of conventional speech recognition in case of disjoint (non-overlapping) speech.

In this article, a model for multi-speaker speech recognition is proposed. We will release code to reproduce our experiments here: https://github.com/cant-access-rediska0123/multi-channel-tranformer. The presented approach works correctly if no more than two people participate in the conversation on the input audio recording, however, the architecture of the presented model can be generalised to the situation of any limited number of speakers. In all other aspects, the problem statement is the most general (including the input speech may overlap).

Existing solutions

Currently, there are several approaches to solving the problem of multi-speaker speech recognition. A detailed overview of the existing solutions of the diarization problem and approaches to multi-speaker speech recognition is presented in [1].

One of the standard problem statements in working with multi-speaker speech is audio mark-up using speaker identifiers. The first methods for multi-speaker speech recognition solve the problem of conventional speech recognition by predicting, together with parts of the text, the unique identifiers of the speakers who uttered this text (for example, [2-5]). Such approaches do not require the use of separate models for the multi-speaker speech recognition, so this problem boils down to determining speaker identifiers for different audio parts. This could be done using particular methods of diarization – determining who spoke on which segment of audio. Such methods do not usually consider situations with overlapping speech (as for example [2]).

Often, in order to solve the problem of overlapping speech, it is proposed to solve a more complex problem: to automatically divide the signal into several audio tracks (channels), each of which refers to a separate person. The signal is then recognized independently on each track using conventional speech recognition models. This approach is used, for example, in [6]. However, this approach has a number of shortcomings. Firstly, it is necessary to additionally solve the problem of signal separation, which may be more difficult than the original problem (it is shown that such a problem is difficult for human hearing, [7]). Secondly, this approach is limited in the ability to work on improving the quality of models by collecting training data. In the presence of a working neural network architecture, further and significant improvement in the quality of most machine learning systems is achieved by collecting and marking up new training data (including using crowd-sourcing, for example, [8, 9]). Finally, in modern machine learning, there is a trend towards integral end-to-end approaches that struggle with the problem of all cascading approaches, when errors of one model affect subsequent ones. For example, later solutions to the problems of diarization, speech recognition, and speech synthesis have an end-to-end architecture [5, 10, 11].

There are also similar modern approaches that solve the problem of overlapping speech in different ways [3–5]. Those approaches use modern state-of-the-art architectures like transformers to achieve best possible results, but recognize speech of different speakers consequently. This process requires

$$\sum_{i=1\dots n} \left| h_i \right|$$

auto-regressive model calls to transcribe *n* text hypotheses $h_1, h_2, ..., h_n$. The runtime of the multi-speaker speech recognition system can be critical in many situations, for example in case of real-time multi-speak-

Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

er speech recognition, when the model needs to be inferenced fast to transcribe texts with minimum possible delay.

In this article, an end-to-end model for multi-speaker speech recognition with possible overlaps is proposed. This model achieves results that can be compared with the results of modern state-of-the-art solutions, but it reduces the runtime by recognizing multiple speakers in parallel, so that only $\max_{i=1,n} |h_i|$ auto-regressive model calls are required to transcribe text hypotheses $h_1, h_2, ..., h_n$. The structure of the presented model is based on the assumption that no more than two people participate in the dialogue, but it can be generalised to the case of any limited number of participants in the dialogue.

The successfully released transformer architecture [12] is now used in the best models that solve a wide variety of machine learning problems, including speech-related problems. The most recent models often use this architecture [5, 13, 14]. This model structure looks promising in such problems, so the presented model is completely based on the transformer architecture.

Metrics

A common metric for evaluating speech recognition models is Word Error Rate (WER), which is defined as the ratio of the minimum number of inserts, deletions and substitutions of words necessary to convert the hypothesis of the model into the text spoken on the audio recording to the number of words in this text:

WER
$$(h,r) = \frac{I(h,r) + D(h,r) + S(h,r)}{|r|}$$
,

where h is the hypothesis of the model about the speech spoken on the audio recording; r is the target text (spoken on the audio recording); I, D, S are the numbers of inserts, deletions and substitutions of words, respectively.

When evaluating the models of multi-speaker speech recognition, the exact correspondence of the speech recognition texts to the spoken texts is unknown, since the order of speakers in the prediction of the model is unknown. Therefore, the most widely used metric is the Concatenated minimum-Permutation Word Error Rate (cpWER, [15]), which generalises the WER metric to the case of several speakers. CpWER is calculated using the following formula:

$$\operatorname{cpWER}(h,r) = \frac{\min\left(I\left(h_{\pi_{i}},r_{i}\right) + D\left(h_{\pi_{i}},r_{i}\right) + S\left(h_{\pi_{i}},r_{i}\right)\right)}{\sum_{i}|r_{i}|},$$

where *h* is several texts; the hypothesis of the model about the speech spoken by each of the participants of the conversation on the audio recording; *r* is the texts spoken by the speakers on the audio recording; *n* is the number of speakers on the audio recording; the number of texts in *r* and in *h*; $\Pi(n)$ is all possible permutations from *n* elements; π is the permutation that determines the correspondence of the texts from the hypothesis of the model *h* to the texts on the audio recording *r*.

At the same time, if the number of people in the prediction of the model does not match their number on the audio recording, the missing people are filled with empty texts.

In this paper, in order to assess the quality of the model, the cpWER metric is used for the case of two speakers. When calculating it, all (two) possible permutations of the target texts were iterated through.

In order to solve this problem, the transformer architecture is used. The model (Fig. 1) is constructed for the case of no more than two speakers on an audio recording, but it can be generalised to the case of any limited number of people. The model is constructed by analogy with the transformer model for speech recognition [13].

Model: multi-channel transformer

Standard methods are used for pre-processing of the input audio recordings and target texts. Similarly to [13], 80-dimensional mel-spectrograms are used to extract features from the audio recording.

In order to convert target texts, a 8000 tokens-size dictionary (sentencepiece, [16]) pre-trained on training data is used. Thus, one target text corresponding to one speaker is divided into several tokens from this dictionary, where the token is one or more letters in a row, which are often found in the training data in this combination.

The architecture of the multi-channel transformer consists of two parts – encoder and auto-regressive decoder. The encoder does not depend on the recognized texts in any way, so any encoder suitable for the speech recognition problem can be used in its place. The experiments used the encoder inspired by the work of [13], consisting of sequentially applied layers of two convolutions, Positional Encoding and ten Transformer Encoder blocks [12]. Additionally, experiments were carried out with the encoder from [17]. As experiments showed, for the stated problem, the most critical part of the model is the decoder and the network training method, and not the specific architecture of the encoder.

The decoder architecture has been modified for the case of multiple texts. In the presented model, two prefixes of prediction texts independently pass through the same Embedding and Positional Encoding blocks. To get rid of the extra dimension (two blocks of features from two speakers) a linear layer is used reducing the number of features by half. The final Transformer Decoder block predicts the distributions of the next tokens.

Instead of one distribution of the next token $p(x_t | x_1, y_1, ..., x_{t-1}, y_{t-1}, e)$ as it would be in a conventional transformer, the decoder predicts two independent distributions of the following tokens for each speaker:

$$p(x_t|x_1, y_1, ..., x_{t-1}, y_{t-1}, e), p(y_t|x_1, y_1, ..., x_{t-1}, y_{t-1}, e),$$

where e is the output of the encoder encoding the audio recording; x, y is the recognition texts of two speakers. In order to ensure the same length of the sequences x and y, the smaller of them is padded with END tokens denoting the end of the sentence.

This paper does not consider the problem of mark-up of recognized text using speaker identifiers. Therefore, in the training data the first text corresponds to the text of the person who started speaking first, and the second text corresponds to the text of the remaining second person.

Experiments were conducted with two decoding variants. In the first variant, experiments had two decoders corresponding to each of the speakers on the audio recording and recognizing the two hypotheses independently. This variant showed results comparable to the results of the current variant. But due to the doubled number of decoders, the number of parameters in such a model increased by ~ 1.5 times, and both decoders had to learn the same information about the identity of the speakers. Therefore, this approach has been replaced with the second variant – with a single decoder that recognizes both texts together.

In total, the final model for recognizing two speakers turned out to have ~ 1.94×10^8 parameters. The single-channel transformer for conventional speech recognition, by analogy with which a multi-channel transformer was built, have approximately the same (~ 1.8×10^8) number of parameters.

Data

Data with speech recordings of one or two speakers were used to train the multi-channel transformer. In the absence of open-source datasets with short recordings of conversations between two people, data from the widespread LibriSpeech speech recognition dataset [18], which includes 960 hours of audiobook recordings, was taken to train the model. Parts of train-clean and train-other data are used in the process of training.



what number are you... twenty two fifty...

Fig. 1. Scheme of multi-channel transformer

There are no dialogues samples in the LibriSpeech data. Therefore, in addition to single-speaker original examples, synthetic data obtained by mixing pairs of audio recordings are used in training the model to recognize two people (Fig. 2). When mixing, both audio recordings are padded with silence to the same size, and then their average is taken. Pairs of audio recordings are selected so that they are spoken by different people, that is, the unique speaker identifiers given in LibriSpeech for each audio recording were different for these two samples.

When generating another synthetic example in the training process, first audio recording is uniformly selected from all the data (train-clean and train-other), then the second audio recording is uniformly selected among all, excluding audio recordings with the same speaker, then these two audio recordings are mixed with indentation. The indentation is selected randomly with a lower bound of 1 second in order to avoid the possibility of incorrectly determining the order of speakers. This streaming method of generating examples greatly increases the size of the training data.

At the training stage data is pre-processed using several augmentations: changing the audio speed with a coefficient chosen randomly from the interval [0.75, 1.25] and augmenting the spectrogram [19].

Similar data is used in [6] to train the model to separate audio for each of the speakers and subsequent speech recognition on each of the resulting channels. However, in this work, such data are used to train the model on signal separation, and not for end-to-end multi-speaker speech recognition. Also in this article, the case of a large number of speakers and long audio recordings is considered, so such data are not suitable for the multi-channel transformer training.

In order to test the models, several datasets were collected from the LibriSpeech test cases (test-clean and test-other): a dataset with one speaker on each audio recording (original data) and mixed pairs of these audio recordings (each audio recording participated in no more than one synthetic example). The overlap size of audio recordings is measured as a percentage of the length of the segment with overlapping audio from the length of the first audio recording. Mixing was carried out for all possible overlaps from 0 % to 100 % in 10 % increments, forming eleven synthetic test datasets.



Fig. 2. Synthetic data for training multi-channel transformer with two speakers

Baseline solutions

In existing articles with different approaches to solving the problem of multi-speaker speech recognition there is no source code or pre-trained models. Therefore, similarly to [4], in order to compare the results of the multi-channel transformer, a baseline solution for multi-speaker speech recognition was built.

Firstly, the "single-speaker transformer" model was built, recognizing speech on each channel with their known separation. It is using a single-channel transformer model for speech recognition. The architecture and hyperparameters of this model were similar to the multi-channel transformer. The single-speaker transformer does not solve the stated problem of multi-speaker speech recognition because it uses generally unknown information about channel separation, but this model can be considered as a lower estimate for the potential quality of the multi-channel transformer, since their architectures are almost the same.

Secondly, the "transformer with channel separation" model was built. In it, similarly to [6], a model for separating the audio signal is used first (specifically, the Sepformer model, [20], pre-trained in [21]), and then the same single-channel transformer for speech recognition is used for each channel.

The single-channel speech recognition transformer and multi-channel transformer were trained on 8 A100 GPUs with a batch size of 16 audio recordings on each GPU. In the training process of the multi-channel transformer, 30 % of the examples were taken uniformly from all audio with one speaker, and 70 % – uniformly from synthetic data with two speakers. In the training of the single-channel transformer, only data with one speaker was used. The multi-channel transformer was trained for eight days, and the single-channel transformer was trained for three days. In both models, the AdamW optimizer ([22]) was used with the parameters weight decay of 0.01, learning rate of 10^{-4} and the subsequent two decreases of the learning rate by ten times when the model error function reached a plateau. Both models used 12 layers of Transformer Encoder, 6 layers of Transformer Decoder, 16 attention-heads, $d_{model} = 512$.

Results

Solutions were tested on test data from mixed pairs of LibriSpeech data. Table 1 shows the results of the cpWER metric for three models: single-speaker transformer, transformer with channel separation, and the presented model. The results are presented for several different overlaps of two audio recordings when they are mixed (0 %, 20, 50, 70, 90 %). The experiments were carried out with transformer models generating recognition using beam search with different search widths (1 and 5).

The results of the single-speaker transformer do not depend on the size of the overlap of the mixed audio recordings, since this model is fed to the input channels of each speaker separately, which are known even before the mixing process and do not depend on its results. The results of the single-speaker transformer can be considered as a lower estimate for the potential quality of the multi-channel transformer, since their architectures are almost the same.

Table 1

CpWER on synthetically mixed data from test-clean and test-other when generating transformer models recognitions using beam search with different search widths (1 and 5) for different overlaps of audio recordings in three compared models: single-speaker transformer (transformer), transformer

with channel separation (sepformer baseline) and the presented model

Synthetic test-clean, cpWER, %						
	Overlap	0	20	50	70	90
	transformer	4.6 ± 0.2				
Beam search, 1	sepformer baseline	21.2 ± 0.4	20.0 ± 0.3	20.2 ± 0.4	20.1 ± 0.2	19.8 ± 0.3
	multi- channel transformer*	5.9 ± 0.4	6.0 ± 0.3	9.8 ± 0.2	12.0 ± 0.3	21.3 ± 0.7
	transformer	4.4 ± 0.2				
Beam search, 2	sepformer baseline	20.6 ± 0.5	19.6 ± 0.5	20.0 ± 0.4	18.6 ± 0.4	18.9 ± 0.7
	multi- channel transformer*	5.7 ± 0.3	5.8 ± 0.4	9.5 ± 0.4	11.5 ± 0.5	20.2 ± 0.7
	transformer	10.6 ± 0.5				
Beam search, 1	sepformer baseline	29.7 ± 0.5	29.5 ± 0.5	29.1 ± 0.6	29.1 ± 0.5	29.5 ± 0.6
	multi- channel transformer*	11.6 ± 0.5	12.3 ± 0.5	16.5 ± 0.5	19.3 ± 0.6	25.5 ± 0.5
Beam search, 2	transformer	10.0 ± 0.4				
	sepformer baseline	29.1 ± 0.6	29.1 ± 0.5	28.3 ± 0.5	29.1 ± 0.5	29.3 ± 0.5
	multi- channel transformer*	11.0 ± 0.4	11.9 ± 0.5	16.0 ± 0.4	18.7 ± 0.5	24.8 ± 0.6

Additionally, Fig. 3 shows a diagram of the dependence of the cpWER metric on different overlaps of audio recordings. To generate the results of this diagram, the model was applied using beam search with a search width of 5.

For synthetic data from test-clean and test-other with overlaps of 0-20 %, the multi-channel transformer shows quality of cpWER ~ 15 % better than the transformer with channel separation, and ~ 1-2 % worse than the transformer with known channel separation. Moreover, it can be seen from Fig. 3 that the multi-channel transformer beats the transformer with channel separation in case of small overlaps (up to 80 % on synthetic test-clean and all overlaps for test-other).

On synthetic audio recordings with a large overlap, the multi-channel transformer tends to poorly, but consonantly, recognize overlapping parts of utterances. An example of such a recognition result is presented in Fig. 4. For an overlap of 90 % in the example presented, the hypothesis of the multi-channel transformer gets $cpWER \sim 40$ % worse than its predictions for an overlap of 10 %. This may be due to the complexity of speech recognition in the case of large overlaps, which turns into a channel separation problem.



Fig. 3. Diagram of the cpWER dependence on the overlap of mixed audio recordings on synthetic data from test-clean and test-other for the three models compared

Multi-channel transformer, 10% overlap				
Text 1	these perverters of the righteousness of christ resist the father and the son and the works of them both			
Hypothesis 1	these proverbs of the righteousness of crise resist the father and the son and the works of them both			
Text 2	i pass away yet i complain and no one hears my voice			
Hypothesis 2	i pass away yet i can t blame and no one here s my voice			
cpWER	22.5%			
Multi-channel transformer, 90% overlap				
Text 1	these perverters of the righteousness of christ resist the father and the son and the works of them both			
Hypothesis 1	these prefer husband outrageways of christ he says the father and the snow and the hoards of the sofa			
Text 2	i pass away yet i complain and no one hears my voice			
Hypothesis 2	i so so way yet i do so frightened the miller appears in my voice			
cpWER	64.5%			
Sepfomer baseline, 90% overlap				
Text 1	these perverters of the righteousness of christ resist the father and the son and the works of them both			
Hypothesis 1	these prefers of the righteousness of christ resists the father and the son and the works of the boat			
Text 2	i pass away yet i complain and no one hears my voice			
Hypothesis 2	it s i passed away yet i can flee and no one hears my voice			
cpWER	29%			

Fig. 4. Recognition of the multi-channel transformer and transformer with channel separation on one example from synthetic test-clean for different overlap sizes. The words correctly recognized by the model are aligned with their originals. The part of the texts that overlaps is highlighted in grey

Sepformer baseline, 90% overlap				
Text 1 Hypothesis 1	well she was better though she had had a bad night			
Text 2	what was the better so she had got it by hight were i but already on the cart			
Hypothesis 2 cpWER	how she was so she gathered but already on the cart 72.2%			
Multi-channel transformer, 20% overlap				
Text 1	well she was better though she had had a bad night			
Hypothesis 1	well she was better though she had had a bad night			
Text 2	were i but already on the cart			
Hypothesis 2 cpWER	or i but all ready on the cards 22.2%			

Fig. 5. Recognition of the transformer with channel separation and multi-channel transformer on one example from synthetic test-clean. The words correctly recognized by the model are aligned with their originals. The part of speech recognition obtained after the artefacts of separating the audio recording into channels is highlighted in grey

On the other hand, the single-speaker transformer tends to recognize multi-speaker speech worse due to various artefacts in the audio recording channels after separation. For example, in the sample from Fig. 5, after splitting into channels, part of the first audio recording was duplicated in a quiet tone on the second channel, which is why both texts from subsequent speech recognition contain recognition of this piece.

Table 2

CpWER in the case of one speaker for test-clean and test-other for the single-speaker (transformer) and multi-channel transformer when using beam search with different search widths (1 and 5)

	Test-clean,	cpWER, %	Test-other, cpWER, %		
	beam search, 1	beam search, 5	beam search, 1	beam search, 5	
Transformer	4.5 ± 0.3	4.3 ± 0.3	10.0 ± 0.3	9.3 ± 0.4	
Multi-channel transformer*	5.3 ± 0.3	4.7 ± 0.3	10.8 ± 0.4	9.8 ± 0.4	

Table 2 shows the results of experiments with the multi-channel transformer and single-channel transformer on audio recordings with one speaker. The results show that in the case of one speaker instead of two, the presented model differs in quality from the usual transformer model that can recognize the speech of only one speaker in only $\sim 1 \%$ cpWER.

The proposed multi-channel transformer model achieves results similar to modern multi-speaker recognition systems (~4–5 % on test-clean single-speaker samples and ~ 4–6 % cpWER on two-speaker synthetic test-clean samples), but has faster inference due to parallel speakers recognition architecture. The number of model runs needed to recognize texts $h_1, h_2, ..., h_n$ is $\max_{i=1...n} |h_i|$. The number of model runs required by modern consequent speaker recognition techniques is $\sum |h_i|$.

Thus, the proposed solution speeds up model inference by two times in the best case of two-speaker speech (when the length of different speaker hypotheses are approximately the same), and has the same speed in the worst case (when only one speaker is presented).

Conclusion

Based on the results of the study, a model was built that solves the problem of possibly overlapping multi-speaker speech recognition of no more than two speakers. The model was trained and tested on synthetic data from LibriSpeech and showed better quality than the transformer with channel separation for small overlaps of audio recordings (up to 80 % cpWER on test-clean and 90 % cpWER on test-other), as well as a small difference from the model that recognizes a single speaker, in the case of such data. The model also showed results similar to modern state-of-the-art multi-speaker speech recognition solutions, but can be inferenced faster, which could be beneficial in many tasks, for example, real-time multi-speaker speech recognition.

The presented model has a potential for many applications, for example, transcription of meeting records, or mark-up and subsequent analysis (for example, [23]) of multi-speaker speech. To solve such problems, it will only be necessary to overcome the problem of the model's inability to recognize speech on sufficiently long audio recordings (this problem, as for any similar auto-regressive models, arises due to the limitations in the memory of computing resources) and, possibly, to investigate generalisations of the model to a larger number of speakers. More details about these problems and their possible solutions can be found in the "Future work" section.

Another result of the study is a data construction scheme. The presented streaming method of generating two-speakers training examples by mixing pairs of audio recordings with different overlaps greatly expands the size of the dataset, and also forces the models to learn to recognize audio recordings with possible speech overlaps.

In addition, it is expected that the multi-channel transformer will work no worse on real audio recordings with two speakers than on synthetic data, since the problem of dividing audio into channels, which in fact the multi-channel transformer is facing in the case of large overlaps in synthetic audio recordings, is more difficult than conventional recognition of dialogue speech. However, due to the lack of open-source datasets with short dialogues, the relevant studies were conducted only on a closed-source dataset and therefore are not presented in this article. The same studies on proprietary data showed that the multi-channel transformer demonstrates high quality not only on LibriSpeech audio recordings recorded with a minimum amount of background noise by speakers in the studio, but also on noisier data.

Future work

One of the shortcomings of the proposed solution is its inability to recognize speech on long audio recordings due to the limitations in the memory of modern computing resources. A possible way to solve this problem is to combine the recognition of small audio recording windows, which may also allow solving the problem of recognizing audio recordings in real time. When applying this approach to the proposed model, it will be necessary to match the local order of speakers in the audio recording window to their global order, which is a problem for further study.

Additionally, a useful functionality of the model would be to mark up the utterances of each speaker in time inside the audio recording, that is, to determine the time intervals for each word/symbol from the recognition hypothesis. Such information, for example, would help when gluing window recognitions of a long audio recording to identify the same words from different windows.

Another field for research is the case of a large number of participants in the dialogue (more than two). In some situations, the upper limit on the number of speakers on an audio recording may be unknown, or it may be very large. It is yet to be found whether the multi-channel transformer generalised to a larger number of speakers can produce the results of the same how high quality.

Another problem is to select the optimal architecture of the proposed model. Experiments were carried out with only two encoder variants (the original and the encoder from [17]) and two decoding variants (with two and one decoder). However, it is not known how much the results could be improved by using any other architecture. Unfortunately, such experiments with the architecture of the model or increasing the upper bound by the number of speakers require a lot of time and computing power.

REFERENCES

1. Park T.J., Kanda N., Dimitriadis D., Han K.J., Watanabe S., Narayanan S. A review of speaker diarization: Recent advances with deep learning. 2021. DOI: https://doi.org/10.48550/arXiv.2101.09624

2. Shafey L.E., Soltau H., Shafran I. Joint speech recognition and speaker diarization via sequence transduction. 2019. DOI: https://doi.org/10.48550/arXiv.1907.05337

3. Kanda N., Horiguchi S., Fujita Y., Xue Y., Nagamatsu K., Watanabe S. Simultaneous speech recognition and speaker diarization for monaural dialogue recordings with target-speaker acoustic models. 2019. DOI: https://doi.org/10.48550/arXiv.1909.08103

4. Kanda N., Gaur Y., Wang X., Meng Z., Chen Z., Zhou T., Yoshioka T. Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers. 2020. DOI: https://doi.org/10.48550/arXiv.2006.10930

5. Kanda N., Ye G., Gaur Y., Wang X., Meng Z., Chen Z., Yoshioka T. End-to-end speaker-attributed ASR with transformer. 2021. DOI: https://doi.org/10.48550/arXiv.2104.02128

Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

6. Raj D., Denisov P., Chen Z., Erdogan H., Huang Z., He M., Watanabe S., Du J., Yoshioka T., Luo Y., Kanda N., Li J., Wisdom S., Hershey J.R. Integration of speech separation, diarization, and recognition for multi-speaker meetings: system description, comparison, and analysis. 2020. DOI: https://doi.org/10.48550/arXiv.2011.02014

7. **Bronkhorst A.** The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica*. 86 (2000): 117–128. DOI: 10.3758/s13414-015-0882-9

8. Pavlichenko N., Stelmakh I., Ustalov D. Crowdspeech and voxdiy: Benchmark datasets for crowdsourced audio transcription. 2021. DOI: https://doi.org/10.48550/arXiv.2107.01091

9. Lyudvichenko V., Vatolin D. Predicting video saliency using crowdsourced mouse-tracking data. 2019. DOI: http://dx.doi.org/10.30987/graphicon-2019-2-127-130

10. Fujita Y., Kanda N., Horiguchi S., Xue Y., Nagamatsu K., Watanabe S. End-to-end neural speaker diarization with self-attention. 2019. DOI: https://doi.org/10.1109/ASRU46091.2019.9003959

11. Kim J., Kong J., Son J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. 2021. DOI: https://doi.org/10.48550/arXiv.2106.06103

12. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. *Attention is all you need*. 2017. DOI: https://doi.org/10.48550/arXiv.1706.03762

13. Mohamed A., Okhonko D., Zettlemoyer L. *Transformers with convolutional context for ASR*. 2020. DOI: https://doi.org/10.48550/arXiv.1904.11660

14. Li N., Liu S., Liu Y., Zhao S., Liu M., Zhou M. Neural speech synthesis with transformer network. 2018. DOI: https://doi.org/10.48550/arXiv.1809.08895

15. Watanabe S., Mandel M., Barker J., Vincen E., Arora A., Chang X., Khudanpur S., Manohar V., Povey D., Raj D., Snyder D., Subramanian A.S., Trmal J., Yair B.B., Boeddeker C., Ni Z., Fujita Y., Horiguchi S., Kanda N., Yoshioka T., Ryant N. Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. 2020. DOI: 10.21437/CHiME.2020-1

16. Kudo T., Richardson J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. 2018. DOI: https://doi.org/10.48550/arXiv.1808.06226

17. Gulati A., Qin J., Chiu C.-C., Parmar N., Zhang Y., Jiahui Yu W.H., Wang S., Zhang Z., Wu Y., Pang R. Conformer: Convolution-augmented transformer for speech recognition. 2020. DOI: 10.21437/Inter-speech.2020-3015

18. Panayotov V., Chen G., Povey D., Khudanpur S. Librispeech: An ASR corpus based on public domain audio books. 2015. DOI: https://doi.org/10.1109/ICASSP.2015.7178964

19. Park D.S., Chan W., Zhang Y., Chiu C.-C., Zoph B., Cubuk E.D., Le Q.V. Specaugment: A simple data augmentation method for automatic speech recognition. 2019. DOI: 10.21437/Interspeech.2019-2680

20. Subakan C., Ravanelli M., Cornell S., Bronzi M., Zhong J. Attention is all you need in speech separation. 2021. DOI: https://doi.org/10.48550/arXiv.2010.13154

21. Ravanelli M., Parcollet T., Plantinga P., Rouhe A., Cornell S., Lugosch L., Subakan C., Dawalatabad N., Heba A., Zhong J., Chou J.-C., Yeh S.-L., Fu S.-W., Liao C.-F., Rastorgueva E., Grondin F., Aris W., Na H., Gao Y., Mori R.D., Bengio Y. Speech-brain: A general-purpose speech toolkit. 2021. DOI: http://dx.doi. org/10.21437/Interspeech.2022-10644

22. Loshchilov I., Hutter F. Decoupled weight decay regularization. 2018. DOI: https://doi.org/10.48550/ arXiv.1711.05101

23. Narayanan S., Georgiou P. Behavioral signal processing: Deriving human behaviorial informatics from speech and language. 2013. DOI: https://doi.org/10.1109/JPROC.2012.2236291

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Фадеева Екатерина Сергеевна Ekaterina S. Fadeeva E-mail: rediska@yandex-team.ru

Ершов Василий Алексеевич Vasily A. Ershov E-mail: noxoomo@yandex-team.ru

Поступила: 29.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023. Submitted: 29.11.2022; Approved: 26.12.2022; Accepted: 12.01.2023. Research article DOI: https://doi.org/10.18721/JCSTCS.15407 UDC 004.852



FIXING 1-BIT ADAM AND 1-BIT LAMB ALGORITHMS

D.A. Tarasov¹ ⊠ , V.A. Ershov² ¹ National Research University Higher School of Economics, St. Petersburg, Russian Federation; ² Yandex LLC, St. Petersburg, Russian Federation

^{III} tarasov.denis.al@gmail.com

Abstract. Today, various neural network models are trained using distributed learning in order to reduce the time spent. The most common way of distributed learning today is the approach, in which the data are divided into parts and sent along with the model to different devices, each device calculates updates for the model, then the updates are aggregated on the server, the server updates the weights of the model and transfers their new version to the devices. Slow network communication between devices can significantly reduce distribution efficiency. Recent studies propose one-bit versions of the Adam and LAMB algorithms, which can significantly reduce the amount of transmitted information, thus improving the scalability of training. However, it turned out that these algorithms diverge in some neural network architectures. The goal of this work is an empirical study of these algorithms, to find the solution of the discovered divergence problem and propose new aspects of testing gradient descent algorithms.

Keywords: machine learning, deep learning, gradient descent, distributed training, optimization

Acknowledgements: We thank Andrey Kirilenko for help in solving different technical issues, which appeared during our research. We also thank Gleb Yengalych for comments on the manuscript.

Citation: Tarasov D.A., Ershov V.A. Fixing 1-bit Adam and 1-bit LAMB algorithms. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 86–97. DOI: 10.18721/JCSTCS.15407

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15407 УДК 004.852



РЕШЕНИЕ ПРОБЛЕМ АЛГОРИТМОВ 1-BIT ADAM И 1-BIT LAMB

Д.А. Тарасов¹ [№] , В.А. Ершов²

¹ Национальный исследовательский университет «Высшая школа экономики», Санкт-Петербургский филиал, Санкт-Петербург, Российская Федерация; ² «Яндекс», Санкт-Петербург, Российская Федерация ⊠ tarasov.denis.al@gmail.com

Аннотация. На сегодняшний день различные нейросетевые модели учат с помощью распределенного обучения, чтобы снизить затрачиваемое время. Самым распространенным способом распределенного обучения является подход, при котором данные разбиваются на части и вместе с моделью отправляются на разные устройства, каждое устройство вычисляет обновления для модели, затем обновления аггрегируются на сервере, сервер обновляет веса модели и передает их новую версию на устройства. Медленное сетевое взаимодействие, связывающее устройства, на которых происходит обучение, может значительно снизить эффективность распределения. Недавние исследования предлагают однобитные версии алгоритмов Adam и LAMB, позволяющие сократить объём передаваемой информации в несколько раз, вследствие чего масштабируемость обучения улучшается. Однако на практике оказалось, что данные алгоритмы расходятся на некоторых архитектурах нейронных сетей. Цель статьи — эмпирическое исследование указанных алгоритмов, решение обнаруженной проблемы расходимости, а также рассмотрение новых аспектов для тестирования алгоритмов градиентного спуска.

Ключевые слова: машинное обучение, глубинное обучение, градиентный спуск, распределенное обучение, оптимизация

Благодарности: Благодарим Андрея Кириленко за помощь в решении технических проблем, возникших во время исследования. Также благодарим Глеба Енгалыча за комментарии к рукописи.

Для цитирования: Tarasov D.A., Ershov V.A. Fixing 1-bit Adam and 1-bit LAMB algorithms // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 86–97. DOI: 10.18721/ JCSTCS.15407

1. Introduction

The latest breakthroughs in deep learning bring about many challenges from areas such as natural language processing, computer vision, and more. The training of neural networks is usually performed on GPUs (Graphical Processing Units) or TPUs (Tensor Processing Units) and their power constantly grows, but the training speed-up is compensated by the fact that models become more complex and the amount of data used for training increases. Training a single model may require weeks, while people need to run many experiments to find a good set of hyperparameters or continuously train existing models using new available data. Therefore, there is a lot of research, the goal of which is to reduce training time while maintaining quality.

Optimization problems arising in training neural network models are solved using the stochastic gradient descent (SGD) method and its various modifications. The aim of several studies is to get an optimization algorithm with not only better convergence rate in theory, but one which provides a speed-up in practice as well: SGD with Nesterov momentum, RMSProp [1], Adagrad [2], Adadelta [3], Adam [4],

LARS [5], LAMB [6], Novograd [7], etc. At the moment, SGD with Nesterov momentum and Adam are the most used ones.

In terms of computation speed, one of the most effective ways to speed up training is to increase the batch size for each training step while reducing the number of steps. Using large batches can lead to worse performance of the trained model, so either some tricks, such as in [8], or suitable optimization methods, such as LARS and LAMB, are required. GPUs and TPUs have limited video memory, so the following approach has found widespread use: the data are divided into the required number of parts and training is performed on several devices, while all the calculated gradients are averaged before the gradient descent step, so that all processes have a model with the same weights. The described approach makes it possible to linearly speed up training if the network communication between devices has sufficient bandwidth and suitable methods are used. However, in practice, network bandwidth may turn out to be a bottleneck that does not allow obtaining a linear increase in the training rate [9].

In [10, 11] 1-bit Adam and 1-bit LAMB algorithms were presented, which propose compressing the transmitted data before synchronisation in order to reduce the amount of transmitted information and, accordingly, accelerate the learning process without a loss of quality. One-bit compression is the most compact among compression approaches, which preserves some useful information about each value. However, in practice, during our experiments, it turned out that these algorithms can lead to divergence for certain models.

Our contributions are as follows:

1. We empirically study 1-bit Adam and 1-bit LAMB algorithms convergence on different tasks.

2. We propose to consider two methods that, to our knowledge, have not been discussed during development of new SGD algorithms.

3. Using these techniques we discover weaknesses of 1-bit Adam and 1-bit LAMB algorithms, as well as a proposing a simple empirical solution to one of the critical problems associated with the original implementations.

2. Previous work

In this section, we will provide a short overview of 1-bit optimizers based on [10]. The idea of the onebit modifications of Adam and LAMB comes from the idea of the one-bit SGD [12]. A single step of the SGD algorithm looks like this:

$$w_{t+1} = w_t - \alpha g_t = w_0 - \alpha \sum_{k=0}^t g_k,$$

where w_t is the model weights at step t; x_t is a training sample at step t; α is the parameter of the learning rate, and if $f_w(x)$ is the objective function, then $g_t = \nabla f_{w_t}(x_t)$, these notations are used further.

If we denote the compression operator by C(x), then the SGD step with compression can be done as follows:

$$w_{t+1} = w_t - \alpha C(g_t) = w_t - \alpha (g_t - \delta_t) = w_0 - \alpha \sum_{k=0}^t g_k + \alpha \sum_{k=0}^t \delta_k$$

 δ_k denotes the error introduced by compression after step k. As it is seen, error accumulates and without its compensation, there is no guarantee of convergence. Therefore, the update is slightly modified so that the error does not accumulate:

$$w_{t+1} = w_t - \alpha C(g_t + \delta_{t-1}) = w_t - \alpha (g_t - \delta_t + \delta_{t-1}) =$$

$$= w_0 - \alpha \sum_{k=0}^{t} g_k + \alpha \sum_{k=0}^{t} (\delta_t - \delta_{t-1}) = w_0 - \alpha \sum_{k=0}^{t} g_k + \alpha \delta_t.$$

If α depends on the step number, the error compensation must look like $C\left(g_t + \frac{\alpha_t}{\alpha_{t-1}}\delta_t\right)$ instead of $C\left(g_t + \delta_{t-1}\right)$.

1-bit Adam. 1-bit Adam is a modification of the widely used Adam algorithm. Adam weights update rules:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_t,$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) (g_t)^2,$$

$$w_{t+1} = w_t - \alpha \frac{m_{t+1}}{\sqrt{v_{t+1} + \varepsilon}},$$

where β_1 , β_2 are numbers between 0 and 1; ϵ is a small constant for numerical stability.

Because of the quadratic dependence of the second momentum (v_k) on gradients, it is impossible to apply error-compensated gradient compression similar to SGD compression.

$$v_{t+1} = \beta_2 v_2 + (1 - \beta_2) (C(g_t + \delta_{t-1}))^2 =$$

= $\beta_2 v_t + (1 - \beta_2) (g_t + \delta_{t-1} - \delta_t)^2 =$
= $\beta_2 v_2 + (1 - \beta_2) ((g_t)^2 + (\delta_{t-1} - \delta_t)^2 + 2g_t (\delta_{t-1} - \delta_t))$

 $(\delta_{t-1} - \delta_t)^2$ from the equality above is not cancelled if the sum is written out, and as a result, the error accumulates.

If we consider $\frac{\alpha}{\sqrt{v_t + \varepsilon}}$ as a changing learning rate (individual for each parameter), then we are forced

to apply compression to the first momentum (m_t) in the following manner:

$$m_{t+1} = C \left(\beta_1 m_1 + (1 - \beta_1) g_1 + \frac{\sqrt{v_{t-1}} + \varepsilon}{\sqrt{v_t} + \varepsilon} \delta_{t-1} \right) =$$

= $\beta_2 v_t + (1 - \beta_2) \left((g_t)^2 + (\delta_{t-1} - \delta_t)^2 + 2g_t (\delta_{t-1} - \delta_t) \right).$

But after the compression, we cannot express v_t and, accordingly, get the coefficient to compensate for the error.

In [10], the authors argued that after some training time, the second momentum stops changing much, and used this observation to build a one-bit version of Adam, see Algorithm 1. First, after several steps (the original work recommends 15-25 % of steps, if the amount for which the model should be trained is known) updates are made in accordance with the original Adam algorithm. After that, the steps begin at which one-bit compression of the first moment occurs, and the second moment remains unchanged. In addition to the compressed vector, its stretching coefficient is also transmitted between training nodes in order to preserve the magnitudes of the initial moments. A theoretical analysis of the algorithm and more details are available in [10].

Algorithm 1:

- 1: **Initialize:** α , β_1 , β_2 , ϵ , w_0 , $m_0 = 0$, $v_0 = 0$, compression errors $\delta = 0$, number of training steps *T*, number of warm up steps T_w , $C_1(x) = \mathbb{1}_{[x \ge 0]}(x) \mathbb{1}_{[x < 0]}(x)$, *n* training nodes
- 2: Perform T_w steps of the original Adam algorithm, keep the last second momentum v_{T_w} and stop updating it

3: **for** $t = T_w, ..., T$ **do**

5: Sample a random subset of examples $x_t^{(i)}$

6:
$$g_t^{(i)} = \nabla_w f_{w_t}(x_t^{(i)})$$

7: $m_t^{(i)} = \beta_1 m_{t-1} + (1 - \beta_1) g_t^{(i)} + \delta_{t-1}^{(i)}$
8: $\hat{m}_t^{(i)} = C_1(m_t^{(i)})$
9: $\delta_t^{(i)} = m_t^{(i)} - \hat{m}_t^{(i)}$
10: (**On server**)
11: $\overline{m}_t' = \frac{1}{n} \sum_{i=0}^n \frac{||m_t^{(i)}||}{||\tilde{m}_t^{(i)}||} \hat{m}_t^{(i)} + \overline{\delta}_{t-1}$
12: $\overline{m}_t = C_1(\overline{m}_t')$
13: $\overline{\delta}_t = \overline{m}_t' - \overline{m}_t$
14: (**On i-th node**)
15: $m_t = \frac{||\overline{m}_t'||}{||\overline{m}_t||} \overline{m}_t$
16: $w_t = w_{t-1} - \alpha \frac{m_t}{\sqrt{v_{T_w} + \epsilon}}$
17: **end for**

1-bit LAMB. Adam, as the research showed, gives poor results with large batch sizes. To solve the problem, the LAMB algorithm was proposed, which takes Adam as a basis and slightly modifies it with the following update rules:

$$\begin{split} m_{t+1}^{(l)} &= \beta_1 m_t^{(l)} + (1 - \beta_1) g_t^{(l)}, \\ v_{t+1}^{(l)} &= \beta_2 v_t^{(l)} + (1 - \beta_2) (g_t^{(l)})^2, \\ u_{t+1}^{(l)} &= \frac{m_{t+1}^{(l)}}{\sqrt{v_{t+1}^{(l)}} + \varepsilon}, \\ c_{t+1}^{(l)} &= c lip \left(\frac{\left\| w_t^{(l)} \right\|}{\left\| u_{t+1}^{(l)} \right\|}, c_{\min}, c_{\max} \right), \\ w_{t+1}^{(l)} &= w_t^{(l)} - \alpha c_{t+1}^{(l)} u_{t+1}^{(l)} \end{split}$$

 c_{\min} , c_{\max} are new algorithm hyperparameters responsible for clipping, index (l) denotes that vector belongs to the l^{th} layer of the network, so coefficient $c_t^{(l)}$ is different for each layer.

In [11], the authors tried to apply the idea from 1-bit Adam without changes, i.e. also stop updating the second moment and use the one calculated at the last step before compression stages. However, this approach led to suboptimal solutions, so the algorithm was complicated to obtain quality results comparable to the original LAMB. We will not describe this algorithm in detail and provide only its pseudocode, see Algorithm 2.

Algorithm 2:

1: **Initialize:** $\alpha, \beta_1, \beta_2, \beta_3, \epsilon, r_{min}, r_{max}, r_{th}, w_0^{(l)}, m_0^{(l)} = 0, v_0^{(l)} = 0, c_{avg}^{(l)} = 0, r^{(l)} = 1$, compression errors $\delta = 0$, number of training steps *T*, number of warm up steps $T_w, C_1(x) = \mathbb{1}_{[x>0]}(x) - \mathbb{1}_{[x<0]}(x), n$ training nodes 2: Perform T_w steps of the original LAMB algorithm, at each step update $c_{avg}^{(l)} = \beta_3 c_{avg}^{(l)} + (1 - \beta_3) c_t^{(l)}$. Keep the last second momentum $v_{T_w}^{(l)}$, stop updating $c_{avg}^{(l)}$. 3: **for** $t = T_w, ..., T$ **do** (On *i*-th node) 4: Sample a random subset of examples $x_t^{(i)}$ 5: $g_t^{(i)} = \nabla_w f_{w_t}(x_t^{(i)})$ 6: $m_t^{(i)} = \beta_1 m_{t-1} + (1 - \beta_1) g_t^{(i)} + \delta_{t-1}^{(i)}$ $\hat{m}_t^{(i)} = C_1(m_t^{(i)})$ $\delta_t^{(i)} = m_t^{(i)} - \hat{m}_t^{(i)}$ 7: 8: 9: (On server) 10: $\overline{m}'_{t} = \frac{1}{n} \sum_{i=0}^{n} \frac{||m_{t}^{(i)}||}{||\hat{m}'^{(i)}||} \hat{m}_{t}^{(i)} + \overline{\delta}_{t-1}$ 11: $\overline{m}_t = C_1\left(\overline{m}_t'\right)$ 12: $\overline{\delta}_t = \overline{m}'_t - \overline{m}_t$ 13: (On *i*-th node) 14: $m_t = \frac{\|\overline{m}_t'\|}{\|\overline{m}_t\|} \overline{m}_t$
for layer *l* do 15: 16: $g_t^{(l)} = \frac{m_t^{(l)} - \beta_1 m_{t-1}^{(l)}}{1 - \beta_1}$ $v_t^{(l)} = \beta_2 v_{t-1}^{(l)} + (1 - \beta_2) (g_t^{(l)})^2$ $r_t^{(l)} = \left\| \frac{v_{t_w}^{(l)}}{v_t^{(l)}} \right\|_{\infty}$ 17: 18: 19: $r_t^{(l)} = \operatorname{clip}(r_t^{(l)}, (1 - r_{th})r_{t-1}^{(l)}, (1 + r_{th})r_{t-1}^{(l)})$ $r_t^{(l)} = \operatorname{clip}(r_t^{(l)}, r_{min}, r_{max})$ $c_t^{(l)} = r_t^{(l)}c_{avg}^{(l)}$ 20: 21: 22: $w_t = w_{t-1} - \alpha c_t^{(l)} \frac{m_t^{(l)}}{\sqrt{v_r^{(l)} + \epsilon}}$ 23: end for 24: 25: end for

3. Discovered effects

During our experiments with the considered algorithms, we discovered effects that are not mentioned in the corresponding works. For research, we used the original implementations provided in the DeepSpeed¹ library. Because 1-bit LAMB in the experiments has the same behaviour and shares the same problem as 1-bit Adam, we show only graphs with 1-bit Adam, in order to reduce the volume.

Divergence. The original work of 1-bit Adam provides the results of training ResNet [13] on ImageNet and CIFAR10 datasets, training and fine-tuning BERT [14], and training DCGAN [15] on CelebFaces dataset. 1-bit LAMB was considered only for the task of training and fine-tuning BERT.

We confirm that training ResNet² using one-bit algorithms on ImageNet and CIFAR10 goes without problems and even find the effect that sometimes the compression stage can be started earlier than recommended by the authors, without loss of quality, which is described below.

We attempted to train networks of the VGG³ [16] family using one-bit algorithms for the classification problem on the CIFAR-10 and ImageNet datasets, Transformer⁴ [17] for the task of translating from English into German on the WMT-14 dataset and Jasper⁵ [18] for speech recognition task on LibriSpeach dataset. From the model listed above, only Jasper did not begin diverging after the onset of the compression stage. VGG divergence with 1-bit Adam is also observed in [19].

Optimization of simple functions. While trying to identify the divergence problem cause, we looked at the behaviour of algorithms when minimising simple functions, such as the sum of squares, and compared it with the behaviour of algorithms without compression.

It can be observed that compression can lead to both slower convergence when approaching the optimum of the function, but at the same time it can have better convergence up to a certain moment, see Fig. 1. This observation shows that one-bit versions of the algorithms require more optimization steps to get as close to the function optimum as the original algorithms.

Early compression without quality loss. For the considered algorithms, the authors recommend starting the compression stage after 15-25 % of the training steps. However, our experiments showed that when training ResNet on the CIFAR10 dataset, compression can be started even after only 3 optimization steps and the quality of the model does not suffer very much, and after 100 the quality is the same as that of the algorithm without compression, see. Fig. 2. In other settings (for example, when training ResNet on the ImageNet dataset), early compression was affecting the quality of the trained model, but this effect may be the subject of further research.

4. Divergence problem

The divergence problem described in Section 3 is critical to the use of the algorithms. Our goal was to identify the cause of this behaviour and find a solution.

Zero gradients are not the only problem. The DeepSpeed documentation, where the algorithm implementations were taken from, recommends that parameter groups known to contain zero gradients should be updated without compression, since one-bit compression encodes zero into a non-zero value, because of which there may be an update of parameters whose gradients are zero. However, we may not know in advance which parameters have a zero gradient. In addition, as our experiments have shown, such parameters are not the only problem. To test the hypothesis that the problem is in zero gradients, we replaced 1-bit compression, with 2-bit compression — positive values of gradients encoded as 1s, negative values encoded as -1s and zeros encoded as 0s. However, even with the 2-bit compression models diverged, which indicates that zero gradients are not the only problem with the considered algorithms.

¹ https://github.com/microsoft/DeepSpeed

² Implementations were taken from https://github.com/pytorch/vision and https://github.com/kuangliu/pytorch-cifar

³ Implementations were taken from https://github.com/pytorch/vision and https://github.com/chengyangfu/pytorch-vgg-cifar10

⁴ Implementation was taken from https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Machine-Translation

⁵ Implementation was taken from https://github.com/NVIDIA/DeepLearningExamples



Fig. 1. Minimising $x^2 + y^2$ function with the original Adam and 1-bit Adam algorithms: *a* – optimization with learning rate 10^{-4} for 500 iterations; *b* – learning rate is 10^{-6} and the number of iterations is set to 7000; "compression *n*" denotes that compression stage started after *n* optimization steps



Fig. 2. Training ResNet on CIFAR10 dataset with usage of Adam and 1-bit Adam algorithms. Each epoch contains 250 optimization steps; "compression n" denotes that compression stage started after n optimization steps; a - loss function values, b - accuracy of prediction on the test set

Different magnitudes of gradients. To understand the possible reason, we decided to analyse the values of the gradients in different models. As it turned out, the ResNet gradients have values of close orders for different layers, the maximum difference in mean and median values is about 100 times, see Fig. 3. At the same time, for example, VGG16 gradients can be very different: the differences of mean and median values are of the 10⁸ order, see Fig. 4. For the Transformer model, mean gradient values are not that representative but median values show that gradient values differ from each other a lot in this model too, see Fig. 5. From these observations, we concluded that this might be the problem.

If we apply the compression procedure used in Algorithm 1 and Algorithm 2 to the following vector of numbers

$$m:=(10^{-4}, 10^{-4}, -10^{-3}, -10^{-2}, -10^{-6}),$$

as a result of compression, there will be a vector

$$\hat{m}$$
: = (1, 1, -1, -1, 1).

After multiplying by scaling coefficient, we end up with vector



Fig. 3. Mean and median values of gradients from different layers of ResNet18 at 1000th training step with Adam optimizer on CIFAR10 dataset



Fig. 4. Mean and median values of gradients from different layers of VGG16 at 1000th training step with Adam optimizer on CIFAR10 dataset



Fig. 5. Mean and median values of gradients from different layers of Transformer at 1000th training step with Adam optimizer on WMT-14 dataset

$$\frac{\|m\|}{\|\hat{m}\|}\hat{m} = 0.0045 \cdot (1, 1, -1, -1, 1).$$

Thus, it turns out that the last element of the restored vector with value of 0.0045 differs from the last element with value of 10^{-6} in the original vector by 4500 times. As it turns out in practice, such changes for the gradient may be enough for the model to start diverging even after one optimization step with compression. If in a vector with zero elements there is at least one nonzero, then after compression and restoration of the norm in the new vector there will be no zeroes. Considering the example with the sum of squares, we can also observe this effect: one-bit optimizers may start diverging if one of the arguments is an order of magnitude larger than the second, see Fig. 6.

Possible solution. As shown above, after the procedure for compressing the vector of the first momenta, some of the first momenta can become orders of magnitude larger than the true ones. Then in the formula of weights update

$$w_t = w_{t-1} - \alpha \frac{m_t}{\sqrt{v_{T_w}} + \varepsilon} \tag{1}$$

too large update step will be made for the corresponding parameters (the denominator is a constant), which may lead to a model divergence.

To solve the problem, we tried several approaches, for example, clip the first momentum at each training node after synchronisation so that the values do not exceed those obtained in the previous step, or restore the second momentum, as is done in line 18 of Algorithm 2, to use it for weights update. Nevertheless, none of this worked and led to a divergence, most likely due to the fact that the changes were not taken into account in the error compensation rules, or did not give a significant effect immediately – one step with problematic compression may be enough for the models to diverge, i.e. the problem must be solved immediately.

There is an option to change the value of the hyperparameter α to a smaller value or increase *e* after the start of compression. This method works, but it requires an additional search for the values, because we want the model trained in this way to show comparable results. In this regard, after starting training with compression, we propose to put the *e* under the square root in (1) so the update rule has the following form:

$$w_t = w_{t-1} - \alpha \frac{m_t}{\sqrt{v_{T_w} + \varepsilon}}.$$
(2)

This modification does not complicate the calculations in any way, and, as our experiments showed, the convergence of unproblematic models does not suffer in any way, and for divergent models, it helps to prevent divergence while saving performance. Fig. 7 shows how this change can help to train VGG16 which diverged with original implementations on the CIFAR10 dataset.

5. Conclusion and future work

In this work, we conducted an empirical study of 1-bit Adam and 1-bit LAMB algorithms and found a problem associated with the original implementation. We proposed a simple solution to the problem: it is computationally simple and quite effective in practice, while not requiring any additional tuning of the parameters. Further studies of the considered algorithms can be focused on their application to other neural network architectures, search for a non-empirical solution to a discovered problem or the effect that in some situations compression can be started almost from the very beginning of model training.

In addition, we hope that the techniques that we used in this work to find problems will be taken into account when creating new modifications of gradient descent where information about the gradients of



Fig. 6. Divergence of 1-bit Adam while minimising $x^2 + y^2$ function because of different magnitude orders of parameters



Fig. 7. VGG16 training on CIFAR10 dataset using Adam and 1-bit Adam modified according to (2). Each epoch contains 24 update steps; "compression *n*" means that compression started after *n* update steps; a - loss function value, b - prediction accuracy on the test set

one parameter used when updating other parameters. Using these techniques can help detect problems similar to those that we found in 1-bit Adam and 1-bit LAMB.

REFERENCES

1. Hinton G., Srivastava N., Swersky K. RMSProp: Divide the gradient by a running average of its recent magnitude. *Neural networks for machine learning*, Coursera lecture 6e, 2012, P. 13.

2. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011, vol. 12 (7).

3. Zeiler M.D. Adadelta: An adaptive learning rate method. arXiv preprint arXiv: 1212.5701, 2012.

4. Kingma D., Ba J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014, 12.

5. You Y., Gitman I., Ginsburg B. Large batch training of convolutional networks. *arXiv preprint arXiv:* 1708.03888, 2017.

6. Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, Cho-Jui Hsieh. Large batch optimization for deep learning: Training Bert in 76 minutes. *arXiv preprint arXiv: 1904.00962, 2019.*

7. Ginsburg B., Castonguay P., Hrinchuk O., Kuchaiev O., Lavrukhin V., Leary R., Li J., Huyen Nguyen, Yang Zhang, Cohen J.M. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. *arXiv preprint arXiv: 1905.11286, 2019.*

8. Goyal P., Dollár P., Girshick R., Noordhuis P., Wesolowski L., Kyrola A., Tulloch A., Yangqing Jia, Kaiming He. Accurate, large minibatch SGD: Training imagenet in 1 hour. *arXiv preprint arXiv: 1706.02677*, 2017.

9. Zhen Zhang, Chaokun Chang, Haibin Lin, Yida Wang, Raman Arora, Xin Jin. Is network the bottleneck of distributed training? *Proceedings of the Workshop on Network Meets AI & ML*, 2020, Pp. 8–13.

10. Hanlin Tang, Shaoduo Gan, Ammar Ahmad Awan, Samyam Rajbhandari, Conglong Li, Xiangru Lian, Ji Liu, Ce Zhang, Yuxiong He. 1-bit Adam: Communication efficient large-scale training with Adam's convergence speed. *ICML*, 2021.

11. Conglong Li, Ammar Ahmad Awan, Hanlin Tang, Samyam Rajbhandari, Yuxiong He. 1-bit LAMB: Communication efficient large-scale large-batch training with LAMB's convergence speed. *ArXiv*, *abs/2104.06069*, *2021*.

12. Frank Seide, Hao Fu, Jasha Droppo, Gang Li, Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. *15th Annual Conference of the International Speech Communication Association*. Citeseer, 2014.

13. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, Pp. 770–778.

14. Devlin J., Ming-Wei Chang, Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv: 1810.04805, 2018.*

15. **Radford A., Metz L., Chintala S.** Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv: 1511.06434, 2015.*

16. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv* preprint arXiv: 1409.1556, 2014.

17. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, Pp. 5998–6008.

18. Li J., Lavrukhin V., Ginsburg B., Leary R., Kuchaiev O., Cohen J.M., Huyen Nguyen, Ravi Teja Gadde. Jasper: An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv: 1904.03288, 2019.*

19. Shaoduo Gan, Xiangru Lian, Rui Wang, Jianbin Chang, Chengjun Liu, Hongmei Shi, Shengzhuo Zhang, Xianghong Li, Tengxu Sun, Jiawei Jiang, Binhang Yuan, Sen Yang, Ji Liu, Ce Zhang. Bagua: Scaling up distributed learning with system relaxations. *ArXiv, abs/2107.01499, 2021*.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Тарасов Денис Алексеевич Denis A. Tarasov E-mail: tarasov.denis.al@gmail.com

Ершов Василий Алексеевич Vasily A. Ershov E-mail: noxoomo@yandex-team.ru

Поступила: 29.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023. Submitted: 29.11.2022; Approved: 26.12.2022; Accepted: 12.01.2023. Research article DOI: https://doi.org/10.18721/JCSTCS.15408 UDC 004.8



ONE-DIMENSIONAL CONVOLUTIONAL LAYERS IN A NEURAL NETWORK FOR WIND SPEED TIME SERIES ANALYSIS

D.N. Kobzarenko¹ ☐ , A.G. Mustafaev², Z.A. Gasanova³, D.S. Magomedova⁴

^{1,2,3,4} Dagestan State University of National Economy, Makhachkala, Russian Federation ⊠ kobzarenko dm@mail.ru

Abstract. Data analysis using neural networks and deep machine learning is one of the current trends in scientific research in various fields. One of the scientific tasks of this direction is the study and prediction of time series using artificial intelligence. The article discusses the results of experiments on adding one-dimensional convolutional layers to a neural network within the framework of the task of classifying meteorological time series data – wind speed. The accuracy of the forecast is shown to increase due to the inclusion of one-dimensional convolutional layers in the model. The increase in accuracy on the test data set for the problem under consideration is about 9.5 %. Several variants of architectures for building a model with one-dimensional convolutional layers and evaluating the accuracy of their classification after machine learning are given. The results obtained allow us to conclude that the use of one-dimensional convolutional layers in the neural network architecture is effective for identifying and predicting a time series of meteorological parameters.

Keywords: neural network, deep machine learning, 1D convolutional layer, wind speed, time series

Citation: Kobzarenko D.N., Mustafaev A.G., Gasanova Z.A., Magomedova D.S. Onedimensional convolutional layers in a neural network for wind speed time series analysis. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 98–107. DOI: 10.18721/ JCSTCS.15408 The Seventh Conference on Software Engineering and Information Management (SEIM-2022)

Научная статья DOI: https://doi.org/10.18721/JCSTCS.15408 УДК 004.8



ОДНОМЕРНЫЕ СВЕРТОЧНЫЕ СЛОИ В НЕЙРОННОЙ СЕТИ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ – СКОРОСТЕЙ ВЕТРА

Д.Н. Кобзаренко¹ ⊠ , А.Г. Мустафаев², 3.А. Гасанова³, Д.С. Магомедова⁴

^{1,2,3,4} Дагестанский государственный университет народного хозяйства, г. Махачкала, Российская Федерация ⊠ kobzarenko dm@mail.ru

Аннотация. Анализ данных с использованием нейронных сетей и глубокого машинного обучения является одним из современных трендов в научных исследованиях в различных областях. Одна из научных задач этого направления — исследование и прогнозирование временных рядов с помощью искусственного интеллекта. В статье рассмотрены результаты экспериментов по добавлению одномерных сверточных слоев в нейронную сеть в рамках задачи классификации данных метеорологических временных рядов скоростей ветра. Показано повышение точности прогноза за счет включения в модель одномерных сверточных слоев. Повышение точности на наборе тестовых данных для рассматриваемой задачи составляет около 9,5 %. Приведены несколько вариантов архитектур для построения модели с одномерными сверточными слоями и оценка точности их классификации после машинного обучения. Полученные результаты позволяют сделать вывод об эффективности применения одномерных сверточных слоев в архитектуре нейронной сети для идентификации и прогнозирования временного ряда метеорологических параметров.

Ключевые слова: нейронная сеть, глубокое машинное обучение, 1D сверточный слой, скорость ветра, временные ряды

Для цитирования: Kobzarenko D.N., Mustafaev A.G., Gasanova Z.A., Magomedova D.S. One-dimensional convolutional layers in a neural network for wind speed time series analysis // Computing, Telecommunications and Control. 2022. T. 15, № 4. C. 98–107. DOI: 10.18721/ JCSTCS.15408

Introduction

In modern scientific researches and developments, artificial intelligence, mostly represented by neural network models, occupies its niche as a tool for solving a group of tasks, including wind energy parameters prediction [1-3]. As a rule, these tasks are regression, forecasting, classification, generating new data based on templates, searching for outliers in data, etc. At the same time, neural network models can be used not only to solve tasks with a classical formulation, where there are initial data and it is required to obtain the final result with the required accuracy, but also to carry out scientific researches that provide answers to questions about the properties of data arrays.

The relevance of research on wind monitoring data is determined by the high interest in the development of renewable energy sources. The main wind power characteristic is the speed. But the wind direction is also an important parameter, since it affects the orientation of the wind turbine and its efficiency during rotation.

To implement the research, we prepared a database based on observations from four meteorological stations located on the territory of the Republic Dagestan (Russian Federation) for the time period of 2011–2020. The names of the stations, "Akhty", "Derbent", "Kochubey", and "Makhachkala", coincide with the names of the corresponding settlements in the region.

Analysis of wind speed and direction time series using neural network models is a continuation of work [4]. The work showed seasonal changes in the frequency characteristics of the meteorological time series in the Coastal Dagestan.

In paper [5], the analysis of meteorological time series (wind speed and direction) was performed using neural network models for the classification task built on the basis of fully connected layers only. This research showed that wind direction time series are classified with almost 100 % accuracy. However, for wind speed time series, the situation is different: in this case, the overall classification accuracy does not reach 70 %. This suggests that the wind speed time series related to the territories of the region that are close in distance contain much less obvious patterns that cannot be captured by a network built only based on fully connected layers.

Use of one-dimensional convolutional layers [6] is a way to improve the performance of a neural network model. One-dimensional convolutional layers are used in a wide variety of tasks from different science areas and technologies [7–12]. Therefore, this paper discusses the results of experiments on use of one-dimensional convolutional layers for the wind speed time series classifying task.

Why classification but not regression? We suppose that performing classification before regression allows us to answer the following questions:

- How well does a neural network recognize a meteorological station by wind parameters?

- What recognition is better: by wind speed data or by wind direction data?
- What data time intervals are optimal for recognition?
- Which meteorological stations are recognized better or worse?

- How are recognition errors correlated with the geographical and relative location of a meteorological station?

The information obtained as a result of the classification task is very important in performing time series prediction based on the same neural network. It significantly expands knowledge about the object of survey – wind speed and wind direction.

We found no publications considering the meteorological time series classifying task in the same way and knew with certainty that such researches had not been performed for our region before. Therefore, we started from our own results.

Toolkit and data preparation

The work used one of the most popular modern tools – the *Keras* library for the *Python* programming language. *Keras* is a high-level add-on to the basic *Tensorflow* neural network library, which allows you to create models on *Python* more quickly than for example *PyTorch* or *Tensorflow* (separately from *Keras*). As a programming environment, the Google Colaboratory cloud resource was used: it provided the ability to do all the work, display and save the results in an Internet browser.

The source data were a set of text files with structured meteorological data. Each text file contained data observations for a month. The frequency of observations was eight times a day in uniform time intervals (hours:minutes): 00:00, 03:00, 06:00, 09:00, 12:00, 18:00, 21:00.

The data covered the time period of 2011–2020. Each meteorological station was represented by a set of 120 files containing information for 10 years. The first engineering challenge was to combine all the data into a single table. A *Python* script was written to solve it. As a result of the script, the source data were converted into the tabular format of the *Pandas* library. For the convenience of extracting information from the table, a temporary index was formed. After conversion, the result table was ready for further work. It was saved into the open CSV file format.

The next step was to prepare a data sample for modeling. The total sample was n-dimensional arrays, where the last dimension determined the number of data blocks. In neural network models, it is accepted to divide the samples into training, verification, and test ones. Usually 80 % is allocated to the training sample and 20 % to the rest. In this case, it was decided to form samples by time periods: the time period

of 2011-2017 was taken for the training sample, 2018 – for the verification sample and 2019-2020 – for the test sample. Such an approach to sampling excludes data mixing, which would inevitably lead to their imbalance (for example, there may be more data for winter or spring in the training sample than for other seasons), which is not acceptable for the considered task.

The next step was to determine the elements of the sample arrays. For this, the concept of the *data block size* was introduced. Consider *the data block size* as a sequence of measurements that fits into a time interval measured in days. For example, a data block size of 3 days means that it contains 3 * 8 = 24 measurements.

The input model data are X – wind speed, and output data are Y – meteorological station index. To reach the best neural network training results, it is usually required to normalize a dataset or to convert it into the one hot encoding (OHE) format. An analysis of the entire database showed that wind speed values were in the range 0–19 m/s. Therefore, to represent wind speed in OHE, it is sufficient to use a vector of 20 elements. The meteorological station index is also converted into OHE of 4 elements (by number of stations). For example, for a data block size of 2 days, the following vector sizes are obtained:

- for X: 20 * 2 * 8 = 320 elements;
- for Y: 4 elements.

Modelling

The current work is based on the results from [5]. In [5], the optimal parameters of the neural network model for meteorological time series were selected. The values were as follows:

- batch size = 40;
- value for dropout layers = 0.1;
- activate function = 'relu';
- learning rate = 0.000005;
- epoch count = 50;
- optimizer = 'Adam';
- loss = 'categorical crossentropy';
- metrics = 'accuracy'.

Since the main purpose of the work is to improve the forecast indicators by using one-dimensional convolutional layers, then all the given basic parameters of the neural network remain unchanged.

The optimal network model based on fully connected layers only, taken from [5], is shown in Fig. 1.

Next, we performed experiments with one-dimensional convolutional layers. To do this, the neural network used not only the one-dimensional convolution function *Conv1D*, but also the functions: *Spatial-Dropout1D* and *MaxPooling1D*. *Keras* also has the *AvaragePooling1D* function, but as experiments showed, it is not suitable for the solving task.

Here you can also introduce the concept of a block for one-dimensional convolution, which consists of a layers' sequence: *SpatialDropout1D*, *Conv1D* and *MaxPooling1D*. Experiments with enumerating parameters in functions showed that the value of the *pool_size* parameter in *MaxPooling1D* should be 2, the value of the *filters* parameter in *Conv1D* should be 10, and the value of *kernel_size* in *Conv1D* should be 5.

This model consists of a sequence of four fully connected blocks. The fully connected block refers to *Dense* and *Dropout* layers' sequence. Further increase in neurons and blocks count does not improve accuracy rates. Note that in the learning process, using the callback, the best result of the arithmetic mean of the accuracy on the verification and test samples is fixed. The best result of the model training is shown in Fig. 2.

Figure 3 shows the neural network model consisting of one convolutional block and two fully connected blocks. Since convolutional layers, unlike fully connected ones, work only with a two-dimensional tensor, it is necessary to use a *Reshape* layer before, and a *Flatten* layer after it.



Fig. 1. The model based on fully connected layers only

```
The best accuracy on validation and test sets = 69.38 %
\ensuremath{\square} Overall accuracy on the test set: 69.58 %
    Station classification accuracy: Akhty 65.0 %
    Errors on stations:
Derbent - 15
    Kochubey - 1
    Makhachkala - 5
     Station classification accuracy: Derbent 68.33 %
    Errors on stations:
    Akhty - 13
     Kochubey - 0
    Makhachkala - 6
     Station classification accuracy: Kochubey 85.0 %
    Errors on stations:
Akhty - 1
    Derbent - 1
    Makhachkala - 7
     Station classification accuracy: Makhachkala 60.0 %
    Errors on stations:
    Akhty - 2
    Derbent - 10
    Kochubey - 12
```

Fig. 2. The best result of the model based on fully connected blocks only training (screenshot from Google Colaboratory notebook)



Fig. 3. The model based on one convolutional block and two fully connected blocks

The best result of training the model based on one convolutional block and two fully connected blocks is shown in Fig. 4. The figure shows that the training result only by adding one convolutional block was improved by more than 3 %. Next, you can experiment with the number of fully connected and convolutional blocks, and see what the accuracy limit of the forecast can be achieved.

Experiments with the number of fully connected and convolutional blocks led to the final version of the neural network model, which had the best learning result. The model consists of a sequence of two convolutional blocks and four fully connected ones, and allows you to get result that is another 2 % better than the previous one (Fig. 5).

All the models discussed above have a sequential architecture. The idea arose to try to design and test models with parallel branches and their subsequent merging through the concatenate layer. Obviously, each branch must have a difference in data processing. In parallel model branches, it is logical to vary the parameters of the *Conv1D* (*filters, kernel_size*) and *MaxPolling1D* (*pool_size*) *layers*.

Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

```
O
   The best accuracy on validation and test sets = 75.21 %
C→
   Overall accuracy on the test set: 72.92 %
                    Station classification accuracy: Akhty 68.33 %
    Errors on stations:
Derbent - 16
    Kochubey - 2
    Makhachkala - 1
    Station classification accuracy: Derbent 81.67 %
    Errors on stations:
    Akhty - 7
    Kochubev - 0
    Makhachkala -
    Station classification accuracy: Kochubey 83.33 %
    Errors on stations:
    Akhty - 1
    Derbent - 1
    Makhachkala - 8
    Station classification accuracy: Makhachkala 58.33 %
    Errors on stations:
    Akhty - 2
    Derbent - 15
    Kochubey - 8
```

Fig. 4. The best result of the model based on one convolutional block and two fully connected blocks training (screenshot from Google Colaboratory notebook)

C	The best accuracy on validation and test sets = 75.83 $\%$
Ŀ	Overall accuracy on the test set: 75.0 %
	Station classification accuracy: Akhty 73.33 % Errors on stations: Derbent - 14 Kochubey - 0 Makhachkala - 2
	Station classification accuracy: Derbent 83.33 % Errors on stations: Akhty - 6 Kochubey - 0 Makhachkala - 4
	Station classification accuracy: Kochubey 80.0 % Errors on stations: Akhty - 1 Derbent - 1 Makhachkala - 10
	Station classification accuracy: Makhachkala 63.33 % Errors on stations: Akhty - 4 Derbent - 12 Kochubey - 6

Fig. 5. The best result of the model based on two convolutional blocks and four fully connected blocks training (screenshot from Google Colaboratory notebook)

As a result of many experiments, the optimal model of the following architecture was adopted. After the initialization and reshaping layer, parallelization into three branches followed, where each branch was a model from Fig. 3 – one convolutional and two fully connected blocks. The branches differed only in the *kernel_size* parameter, which changed with values 3, 5 and 7. After merging the branches in the *Concatenate* layer, a sequence of four fully connected blocks was added to the model, completely in accordance with the model in Fig. 1.

All the completed it is possible to improve the learning result by more than 3 % (Fig. 6).

In addition, the results of classification accuracy by objects turned out to be more balanced than in previous models. This is especially noticeable if we summarize the learning results for the four considered models in a single table (Table 1).

Table 1 shows the effectiveness of using one-dimensional convolutional layers in parallel branches and varying the *kernel_size* parameter in the *Conv1D* layer.

```
The best accuracy on validation and test sets = 79.17 %
O
C→ Overall accuracy on the test set: 78.33 %
    Station classification accuracy: Akhty 76.67 %
    Errors on stations:
    Derbent - 9
    Kochubey - 0
    Makhachkala - 5
    Station classification accuracy: Derbent 78.33 %
    Errors on stations:
    Akhty - 8
    Kochubey - 0
    Makhachkala - 5
    Station classification accuracy: Kochubey 85.0 %
    Errors on stations:
    Akhty - 1
    Derbent - 0
    Makhachkala - 8
    Station classification accuracy: Makhachkala 73.33 %
    Errors on stations:
    Akhty - 0
    Derbent - 10
    Kochubey - 6
```

Fig. 6. The best result of the model based on tree parallel brunches training (screenshot from Google Colaboratory notebook)

Table 1

Comparison of the model accuracies, %

	M1	M2	M3	M4
OVERALL	69.58	72.92	75.0	78.33
Akhty	65.0	68.33	73.33	76.67
Derbent	68.33	81.67	83.33	78.33
Kochubey	85.0	83.33	80.0	85.0
Makhachkala	60.0	58.33	63.33	77.33

Conclusion

As can be seen from the results of the presented work (Table 1), when processing the wind speed time series, the use of one-dimensional convolutional networks significantly improves the accuracy of the forecast on the test data set. In addition, a model architecture with parallel branches and variation of the convolution kernel size can be effective and more beneficial in the classification task not only due to the accuracy of the overall forecast, but also thanks to balancing the accuracy of the objects forecast.

As well known, neural network model architectures are directly dependent on the source datasets. It would be incorrect to compare our results with the simulation results for other regions, since the climate is different everywhere, while datasets may differ in data size and completeness.

In the paper, we do not offer a template for solving similar tasks, but a solution to improve the neural network accuracy by adding 1D-convolutional blocks, while also providing some experiments with the possible model architecture variants.

REFERENCES

1. Catalao J.P.S., Pousinho H.M.I., Mendes V.M.F. Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable Energy*, 2011, No. 36, Pp. 1245–1251. https://doi.org/10.1016/j.renene.2010.09.016

2. **Paramasivan S.K., Lopez D.** Forecasting of wind speed using feature selection and neural networks. *International Journal of Renewable Energy Research*, 2016, vol. 6 (3), Pp. 833–837. https://doi.org/10.20508/ijrer.v6i3.3855.g6866

3. Doucoure B., Agbossou K., Cardenas A. Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *Renewable Energy*, 2016, No. 92, Pp. 202–211. http://dx.doi.org/10.1016/j.renene.2016.02.003

4. Kobzarenko D.N., Kamilova A.M., Pashtaev B.D. Issledovanie sezonnyh izmenenij chastotnyh harakteristik skorosti i napravleniya vetra v Primorskom Dagestane [Research the seasonal changes in frequency characteristics of wind speed and direction in Coastal Dagestan]. *YUg Rossii: Ekologiya, Razvitie*, 2020, vol. 15, No. 4, Pp. 152–160. (rus). https://doi.org/10.18470/1992-1098-2020-4-152-160

5. **Kobzarenko D.N.** Analiz vremennyh ryadov – skorostej i napravlenij vetra s pomoshch'yu modelej nejronnyh setej i zadachi klassifikacii [Analysis of time series – wind speeds and wind directions using neural network models and classification task]. *Morskie Intellektual'nye Tekhnologii*, 2021. vol. 1., No. 4 (54), Pp. 127–133. (rus). DOI: 10.37220/MIT.2021.54.4.043

6. **Kiranyaz S., Avci O., Abdeljaber O.** 1D convolutional neural networks and applications: A survey. *arXiv* preprint. https://arxiv.org/abs/1905.03554 (2019)

7. Jiao Y., Li N., Mao X., Yao G., Zhao Y., Huang L. (2021) Pulse recognition of cardiovascular disease patients based on one-dimensional convolutional neural network. *Bio-Inspired Computing: Theories and Applications. BIC-TA 2020. Communications in Computer and Information Science*, Springer, Singapore, 2021, vol. 1363. https://doi.org/10.1007/978-981-16-1354-8_20

8. Gao C., Wang J., Zhou X., Xiao F., Ma Q. Classification of lightning electric field waveform based on deep residual one-dimensional convolutional network. *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. ICNC-FSKD 2020. Lecture Notes on Data Engineering and Communications Technologies*, Springer, Cham, 2021, vol. 88. https://doi.org/10.1007/978-3-030-70665-4_179

9. Xu W., Li H. Bearing fault diagnosis based on hermitian wavelet and one-dimensional convolutional neural network. *Advances in Intelligent Automation and Soft Computing. IASC 2021. Lecture Notes on Data Engineering and Communications Technologies*, Springer, Cham, 2022, vol. 80. https://doi.org/10.1007/978-3-030-81007-8_41

10. Yu J., Zhang C., Wang S. Multichannel one-dimensional convolutional neural network-based feature learning for fault diagnosis of industrial processes. *Neural Comput & Applic*, 2021, No. 33, Pp. 3085–3104. https://doi.org/10.1007/s00521-020-05171-4

11. Sujanaa J., Palanivel S., Balasubramanian M. Emotion recognition using support vector machine and one-dimensional convolutional neural network. *Multimed Tools Appl*, 2021, No. 80, Pp. 27171–27185. https://doi.org/10.1007/s11042-021-11041-5

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Кобзаренко Дмитрий Николаевич Dmitry N. Kobzarenko E-mail: kobzarenko_dm@mail.ru

Мустафаев Арслан Гасанович Arslan G. Mustafaev E-mail: arslan_mustafaev@mail.ru

Гасанова Зарема Ахмедовна Zarema A. Gasanova E-mail: cudakharka@yandex.ru **Магомедова Динара Сахратулаевна Dinara S. Magomedova** E-mail: mdc-101085@mail.ru

Поступила: 30.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023. Submitted: 30.11.2022; Approved: 26.12.2022; Accepted: 12.01.2023.