



## ИНФОРМАТИКА

Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2023. Т. 23, вып. 1. С. 113–125  
*Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2023, vol. 23, iss. 1, pp. 113–125

[mmi.sgu.ru](http://mmi.sgu.ru)

<https://doi.org/10.18500/1816-9791-2023-23-1-113-125>

EDN: QQEIGU

Научная статья  
УДК 004.89

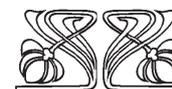
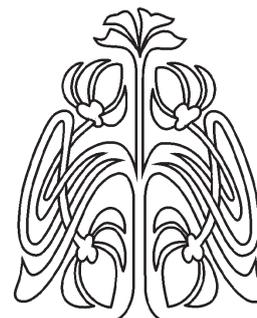
### Метод автоматического поиска структуры и параметров нейронных сетей для решения задач обработки информации

А. Д. Обухов

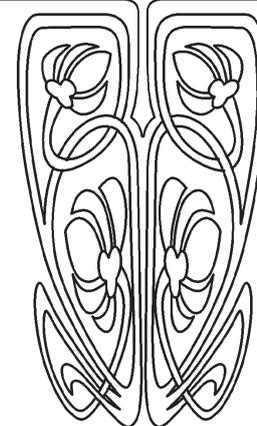
Тамбовский государственный технический университет, Россия, 392000, г. Тамбов, ул. Советская, д. 106

**Обухов Артем Дмитриевич**, кандидат технических наук, доцент кафедры систем автоматизированной поддержки принятия решений, [obuhov.art@gmail.com](mailto:obuhov.art@gmail.com), <https://orcid.org/0000-0002-3450-5213>, AuthorID: 751895

**Аннотация.** Нейронные сети активно применяются при решении различных прикладных задач анализа, обработки и генерации данных. При их использовании одним из сложных этапов является подбор структуры и параметров нейронных сетей (количество и типы слоев нейронов, функции активации, оптимизаторы и т. д.), обеспечивающих наибольшую точность и, следовательно, успешность решения поставленной задачи. В настоящее время данный вопрос решается путем аналитического подбора архитектуры нейронной сети исследователем или разработчиком программного обеспечения. Существующие автоматические инструменты (AutoKeras, AutoGAN, AutoSklearn, DEvol и др.) недостаточно универсальны и функциональны. Поэтому в рамках данной работы рассматривается метод автоматического поиска структуры и параметров нейронных сетей различного типа (многослойных плотных, сверточных, генеративно-состязательных, автоэнкодеров и др.) для решения широкого класса задач. Представлена формализация метода и его основные этапы. Рассмотрена апробация метода, доказывающая его эффективность относительно аналитического решения при подборе архитектуры нейронной сети. Проведено сравнение метода с существующими аналогами, выявлено его преимущество по точности сформированных нейронных сетей и времени поиска



Научный  
отдел





решения. Результаты исследования могут использоваться при решении большого класса задач обработки данных, для которых требуется автоматизировать подбор структуры и параметров нейронной сети.

**Ключевые слова:** нейронные сети, машинное обучение, оптимизация структуры нейронных сетей, анализ и обработка данных, сверточные нейронные сети, автоэнкодеры, генеративно-состязательные сети

**Благодарности:** Работа выполнена при поддержке лаборатории медицинских VR тренажерных систем для обучения, диагностики и реабилитации, Тамбовский государственный технический университет.

**Для цитирования:** Обухов А. Д. Метод автоматического поиска структуры и параметров нейронных сетей для решения задач обработки информации // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2023. Т. 23, вып. 1. С. 113–125. <https://doi.org/10.18500/1816-9791-2023-23-1-113-125>, EDN: QQEIGU

Статья опубликована на условиях лицензии Creative Commons Attribution 4.0 International (CC-BY 4.0)

Article

## Method of automatic search for the structure and parameters of neural networks for solving information processing problems

A. D. Obukhov

Tambov State Technical University, 106 Sovetskaya St., Tambov 392000, Russia

Artem D. Obukhov, obuhov.art@gmail.com, <https://orcid.org/0000-0002-3450-5213>, AuthorID: 751895

**Abstract.** Neural networks are actively used in solving various applied problems of data analysis, processing and generation. When using them, one of the difficult stages is the selection of the structure and parameters of neural networks (the number and types of layers of neurons, activation functions, optimizers, and so on) that provide the greatest accuracy and, therefore, the success of solving the problem. Currently, this issue is being solved by analytical selection of the neural network architecture by a researcher or software developer. Existing automatic tools (AutoKeras, AutoGAN, AutoSklearn, DEvol and others) are not universal and functional enough. Therefore, within the framework of this work, a method of automatic search for the structure and parameters of neural networks of various types (multilayer dense, convolutional, generative-adversarial, autoencoders, and others) is considered for solving a wide class of problems. The formalization of the method and its main stages are presented. The approbation of the method is considered, which proves its effectiveness in relation to the analytical solution in the selection of the architecture of the neural network. A comparison of the method with existing analogues is carried out, its advantage is revealed in terms of the accuracy of the formed neural networks and the time to find a solution. The research results can be used to solve a large class of data processing problems for which it is required to automate the selection of the structure and parameters of a neural network.

**Keywords:** neural networks, machine learning, optimization of the neural network structure, data analysis and processing, convolutional neural networks, autoencoders, generative-adversarial networks

**Acknowledgements:** This work was supported by the Laboratory of medical VR simulation systems for training, diagnostics and rehabilitation, Tambov State Technical University.



**For citation:** Obukhov A. D. Method of automatic search for the structure and parameters of neural networks for solving information processing problems. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2023, vol. 23, iss. 1, pp. 113–125 (in Russian). <https://doi.org/10.18500/1816-9791-2023-23-1-113-125>, EDN: QQEIGU

This is an open access article distributed under the terms of Creative Commons Attribution 4.0 International License (CC-BY 4.0)

## Введение

Методы машинного обучения, в частности нейронные сети, широко применяются при решении задач анализа, обработки и генерации данных в различных сферах человеческой деятельности для автоматизации процессов работы с информацией [1]. При их использовании этап подбора структуры и параметров нейронных сетей является наиболее важным и сложным, так как от него зависит итоговая точность сети и, следовательно, успешность решения поставленной задачи. Изначально выбор типа нейронной сети и поиск ее оптимальных настроек производился аналитическим (экспертным) путем, однако в последние годы появились теоретические методы [2, 3] и программные решения (AutoKeras [4], DEvol [5, 6], AutoSklearn [7], Google AutoML [8], AutoGAN [9]), направленные на автоматизацию данного процесса.

Подходы к автоматическому поиску структур и параметров нейронных сетей связаны с формированием различных конфигураций сетей и их сравнением на контрольном наборе данных, что приводит к значительному потреблению вычислительных ресурсов или временным затратам и, следовательно, затрудняет их широкое распространение и применение. Поиск оптимальных параметров и архитектур аналитическим методом не гарантирует оптимального решения при низкой квалификации разработчиков или исследователей. Следовательно, это ограничивает возможность применения подобных подходов в небольших командах без привлечения специалистов в области машинного обучения и обработки информации с соответствующим уровнем компетенций.

Поэтому актуальной задачей является автоматизация процесса подбора структуры и параметров нейронных сетей. В работе коллектива ученых [10] рассматривается подход, основанный на исследовании пространства возможных архитектур, образованного путем добавления или удаления слоев и изменения их параметров. Для поиска оптимальных настроек сети используются контроллер и функция вознаграждения, оценивающая полученные решения.

Другим теоретическим подходом, направленным на снижение сложности процесса выбора структуры и обучения нейронных сетей, стала методология, предложенная в исследовании [11]. Ее особенностью является реализация единственной, универсальной нейронной сети, способной работать с множеством сценариев за счет использования сотен или тысяч подсетей с одним общим ядром. Это позволяет не искать специализированную структуру для каждой конкретной задачи, а использовать универсальную модель. Эффективность решения растет при росте количества сценариев (задач).

Анализ практических подходов к автоматизированному выбору структуры нейронных сетей позволил выделить следующие программные решения [12]: AutoKeras [4], DEvol [5, 6], AutoSklearn [7], Google AutoML [8], AutoGAN [9]. В каждом из инструментов используются свои подходы для нахождения оптимальных настроек и архитектур нейронных сетей: байесовская оптимизация параметров сетей (AutoSklearn, AutoKeras), генетическое программирование (DEvol), контроллеры на основе нейронных сетей (Google AutoML, AutoGAN).



Проведенный анализ данных инструментов показал следующие результаты, представленные в таблице.

Сравнение существующих подходов по автоматического поиску архитектур нейронных сетей  
Table. Comparison of the existing approaches for automatic search for neural network architectures

Критерий	AutoKeras	DEvol	AutoSklearn	AutoGAN
Классификация данных	+	+	+	–
Регрессия	+	+	+	–
Генерация данных заданного класса	–	–	–	+
Сжатие состояния объекта	–	–	–	–
Прогнозирование данных	+	+	+	–
Восстановление значений	+	+	+	–
Преобразование данных	+	+	–	+
Определение категории задач	+	–	+	–
Подбор структуры нейронной сети	+	+	+	+
Многомерность входных и выходных векторов	–	+	–	+

В качестве основных критериев сравнения использовались функциональные особенности (возможность решения задач классификации, регрессии, генерации, сжатия, прогнозирования, восстановления, преобразования), а также способность автоматического определения категории задач для более точного выбора архитектуры (например, сверточных слоев при работе с изображениями), перебор различных типов нейронных сетей и возможность работы с многомерными входными и выходными данными. Сравнение проведено на основе данных из документации рассматриваемых инструментов [4–9], а также в ходе их установки и проверки функциональности в рассматриваемых категориях задач.

Проведенный анализ позволил сделать следующие выводы. AutoKeras не поддерживает многомерные входные данные, а только одномерный вектор. Выходные данные могут быть представлены единственным значением, что приводит к необходимости создания нескольких выходов для представления векторов. Это создает значительные трудности при обработке данных и ограничивает применимость библиотеки без внесения существенных корректив в структуру исходных данных.

DEvol требует ручной настройки функции потерь, иначе понижается эффективность библиотеки при решении задач регрессии.

AutoSklearn также не поддерживает многомерные входные и выходные данные. Кроме того, невозможно формирование нескольких независимых входов или выходов.

Вышеперечисленные библиотеки не поддерживают работу с нейронными сетями генеративно-состязательного типа и автоэнкодерами.

В отличие от них библиотека AutoGAN реализует подбор структуры генеративно-состязательной сети. Она показывает высокую эффективность при генерации данных на основе датасетов CIFAR10 и STL-10, однако ее архитектура крайне тяжело адаптируется под другие наборы данных. Особенно это касается данных, представленных в виде многомерных численных значений, а не изображений, так как средства оптимизации структуры нейронной сети в данной библиотеке основаны на метрике Inception Score и использовании модели Inception V3. Это значительно ограничивает применение этой библиотеки.

Таким образом, на основе проведенного анализа существующих подходов к поиску архитектур нейронных сетей получено, что универсального метода, применимого



для поиска структур и параметров различных нейронных сетей, не существует, что подтверждает актуальность проводимых исследований. В рамках работы планируется реализация и апробация метода, позволяющего реализовать все функциональные критерии, перечисленные в таблице.

## 1. Материалы и методы

Определим структуру и параметры произвольной нейронной сети  $NN$  следующим выражением:

$$NN = (X, Y, W, TN, SN, HN), \quad (1)$$

где  $X, Y$  — набор входных и выходных векторов данных соответственно, размерность  $X$  составляет  $N_X$ ,  $Y$  —  $N_Y$ ;  $W$  — множество всех весов нейронной сети;  $TN$  — тип архитектуры нейронной сети (многослойная плотная, сверточная, генеративно-состязательная, рекуррентная, автоэнкодер и т. д.);  $SN$  — структура нейронной сети, отражающая количество слоев, нейронов в каждом из них, связи между слоями, выбранные функции активации;  $HN$  — гиперпараметры нейронной сети (скорость обучения, размер пакета, параметры регуляризации).

Функция  $Y = NN(X, W)$  математически выражает нейронную сеть  $NN$ . Нахождение весов  $W$  и метода отображения  $X$  и  $W$  в  $Y$  определяет структуру  $SN$ , тип  $TN$  и параметры  $HN$  нейронной сети  $NN$  [13].

Тогда необходимо сформировать вектор выходных данных  $Y = (y_1, \dots, y_m)$  заданной размерности со структурой данных  $SY$  на основе обработки вектора входных данных  $X = (x_1, \dots, x_n)$  заданной размерности со структурой данных с помощью некоторой нейронной сети  $NN_j$ , входящей в множество всех возможных архитектур нейронных сетей  $NN$ . Для решения поставленной задачи в автоматическом режиме необходимо автоматизировать процесс подбора структуры и параметров нейронных сетей в зависимости от категории  $K_i$  задачи и формата исходных данных. Под понятием «структура данных» будем понимать класс данных, однозначно определяющий множество характеристик и значений каждого элемента класса, а также множество операций, которые могут быть к ним применены, и пусть  $S$  — множество всех классов данных. Тогда под структурой данных  $SA$  некоторого вектора  $A = (a_1, \dots, a_t)$  будем понимать образ множества  $A$  при отображении  $fs : A \rightarrow SA$ , где  $SA \in S$ .

Рассмотрим формализованное представление метода автоматического поиска структуры и параметров нейронной сети, включающее следующие этапы.

1. Формализация вектора входных  $X = (x_1, \dots, x_n)$  и выходных  $Y = (y_1, \dots, y_m)$  данных, определение их структур (классов данных):  $SX$  и  $SY$  соответственно.
2. Анализ связей между входными и выходными данными, что позволит определить категорию  $K_i$  решаемой задачи.
3. Подготовка и обработка исходных данных. На этом этапе осуществляется преобразование исходных данных  $X$  путем выполнения различных операций: нормализации, округления, перехода от непрерывных значений к категориальным, использование разреженных матриц вместо дискретных значений, лемматизация, очистка исходных данных от лишней или служебной информации и т. д. Реализация перечисленных операций возможна в автоматическом режиме за счет набора библиотек программного кода, а их применение позволяет ускорить работу нейронных сетей и повысить их точность. Каждая категория задач  $K_i$  реализует свою процедуру обработки данных.
4. В соответствии с выбранной задачей  $K_i$  в автоматическом режиме выбирается архитектура (1) нейронной сети  $NN^* \in NN$ : итерационно осуществляется



анализ точности нейронных сетей  $NN_j \in NN$  на небольшой выборке данных  $X_0 \in X$ . К основным варьируемым параметрам относятся:

- архитектура модели нейронной сети (последовательная, функциональная);
- тип используемых слоев нейронов (плотные для численных значений, сверточные для изображений, рекуррентные для последовательностей, слои нормализации для уменьшения эффекта переобучения);
- количество плотных слоев нейронов (от 1 до 20);
- количество слоев исключения (dropout) (от 0 до 9);
- количество специализированных слоев (сверточных, рекуррентных и т.д.) (от 1 до 10);
- количество нейронов в слое (от 5 до 2000);
- тип активационных функций (линейная, сигмоидальная, гиперболический тангенс, ReLU, SELU);
- оптимизатор (SGD, Adam, Adamax, Nadam, RMSprop);
- функция потерь (MSE, MAE, MAPE, кросс-энтропия, бинарная энтропия);
- количество эпох обучения (от 3 до 300).

Подмножество сетей  $NN_a \subset NN$ , показывающее наилучшую точность, используется дальше и модифицируется, подмножество  $NN \setminus NN_a$  отбрасывается. Этап 4 повторяется до тех пор, пока не будет достигнута требуемая точность либо разрешенное время на выполнение этапа. Для каждой задачи  $K_i$  определяется конкретное подмножество возможных архитектур и параметров нейронных сетей, что позволяет сократить время поиска оптимальной архитектуры и исключить изначально неэффективные решения. Если структура входных данных соответствует изображениям (определяется на шаге 2 и 3), то используются сверточные слои, если многомерным векторам временных последовательностей — рекуррентные и т.д. Таким образом, в рамках каждой категории задачи возможно сократить максимальное количество возможных сочетаний параметров и структуры нейронной сети с сотен миллионов до нескольких десятков тысяч.

5. Нейронная сеть  $NN^*$  обучается на всем тренировочном наборе  $X_{train} \in X$ . Оценивается ее точность  $A$ . В случае превышения ошибки  $A(NN^*)$  заданного порога  $\varepsilon$  осуществляется возврат к этапу 4. Способ вычисления ошибки (например, на основе среднеквадратичного отклонения, перекрестной энтропии или иных метрик) определен для каждой задачи.
6. Если обученная нейронная сеть обеспечивает требуемую точность ( $A(NN^*) \leq \varepsilon$ ), то осуществляется контрольный замер на данных  $X_{test} \in X$ ,  $X_{test} \cap X_{train} = \emptyset$ . Полученные средствами нейронной сети значения  $Y_{test}$  оцениваются на принадлежность области допустимых значений  $Y$ .
7. Нейронная сеть используется для решения выбранной задачи  $K_i$ .

Таким образом, разработанный метод позволяет в автоматическом режиме подобрать необходимые параметры и структуру нейронных сетей в зависимости от выбранной категории задач  $K_i$ .

## 2. Программная реализация метода

Для программной реализации метода использовались язык программирования Python и библиотека по работе с машинным обучением Keras. Выбор языка программирования обусловлен возможностью объективного сравнения с альтернативными решениями, которые также реализованы на Python, наличием удобных инструментов для анализа и обработки больших объемов данных. Необходимо отметить, что пред-



лагаемый метод может быть реализован с применением как различных фреймворков (Caffe, PyTorch, TensorFlow, Darknet, XGBoost и др.), так и языков программирования. Keras использовался по причине простоты реализации нейронных сетей за счет высокого уровня абстракции. Архитектура программного обеспечения имеет следующий вид (рис. 1): первый компонент определяет конкретную категорию задач обработки данных для исходных структур информации, а второй – реализует все основные этапы метода автоматического поиска структур и параметров нейронных сетей.

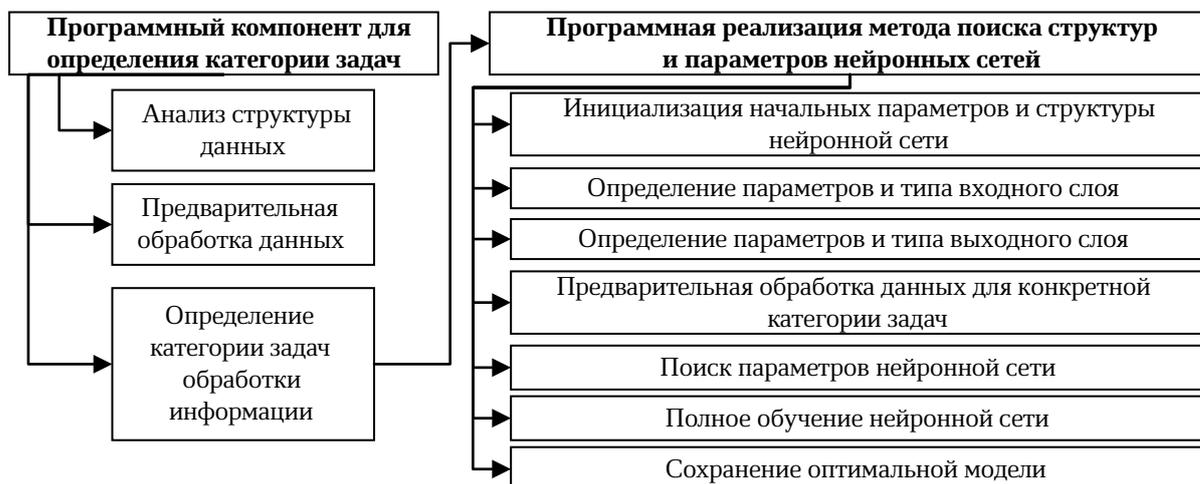


Рис. 1. Структура программного обеспечения для поиска параметров нейронных сетей

Fig. 1. The structure of software for searching the parameters of neural networks

В рамках второго компонента разработан родительский программный класс, обеспечивающий основную функциональность метода, от которого уже осуществлено наследование набора классов, решающих конкретные задачи обработки данных (сжатие данных, прогнозирование, генерация и т. д.).

### 3. Апробация метода

Для проверки метода проведем несколько экспериментов, в которых сравним точность работы нейронных сетей, полученных аналитическим способом, со сформированными разработанным программным обеспечением, реализующим метод автоматического поиска.

В первом эксперименте осуществим подбор структуры и параметров нейронной сети автоэнкодера. В качестве датасета используем MNIST (60 000 изображений размером  $28 \times 28$  пикселей для обучения, 10 000 — для тестирования), так как на нем можно визуально оценить качество сжатия и восстановления из скрытого представления исходных данных. На рис. 2 представлены результаты данного эксперимента, а также значения основных метрик и параметров:  $H$  — длина сжатого состояния;  $T$  — время поиска структуры и параметров нейронной сети;  $L_0, L$  — значение функции потерь (среднеквадратичной ошибки) до и после применения метода.

Во втором эксперименте найдем оптимальную структуру и параметры для генеративно-сопоставительной сети на примере датасета MNIST (рис. 3). Задача усложняется тем, что процесс обучения сетей такого типа нестабилен, что связано с одновременной динамической тренировкой двух конкурирующих моделей. Поэтому важен правильный контроль процесса обучения двух моделей и его своевременный перезапуск в



	Исходный объект	Аналитическое решение	Разработанный метод
$N=10$ $T=478$ $L0=0.15$ $L=0.11$			

Рис. 2. Результаты поиска структуры нейронной сети автоэнкодера

Fig. 2. Results of searching for the structure of an autoencoder neural network

случае дисбаланса в сторону одной из моделей. Для этого используется постоянная оценка качества сгенерированных моделей с применением разработанной ранее модифицированной метрики Inception Score [14]. Особенность метрики заключается в переходе от сети Inception V3 к произвольному классификатору при расчете предельной вероятности (среднего значения условных вероятностей для объектов в группе). Так как задано  $N = 10$  классов цифр, то обучается нейронная сеть, осуществляющая соответствующую классификацию по  $N$  категориям. Для изображений в классификаторе используется последовательность сверточных слоев. Далее эта нейронная сеть подставляется вместо Inception V3 в программную реализацию метрики Inception Score.

	Аналитическое решение	Разработанный метод
$T=1437$ $IS=7.74$		
	$IS=5.78$	

Рис. 3. Результаты поиска структуры генеративно-сопоставительной сети

Fig. 3. Search results for the structure of a generative-adversarial network

Прирост модифицированной метрики Inception Score после применения метода составил 34%. Полученные значения можно интерпретировать следующим образом: аналитическое решение способно сгенерировать 5.78 различных классов из 10, после применения метода — 7.74 из 10 возможных.

В результате в первом эксперименте значение функции потерь после поиска оптимальных параметров и структуры нейронной сети было сокращено на 26%, а во втором эксперименте значение метрики Inception Score улучшено на 34%, что подтверждает применимость изложенного метода для решения поставленной задачи.

#### 4. Сравнительный анализ эффективности метода

Для проверки эффективности предлагаемого метода относительно существующих подходов к автоматическому формированию нейронных сетей осуществим его сравнение с библиотеками AutoKeras, DEvol, AutoSklearn. AutoGAN не участвует в сравнении, так как его возможности ограничены только генерацией изображений.

Разработанный метод отвечает всем критериям, представленным в таблице, что отражает его большую универсальность и функциональность, применимость для



поиска структур и параметров широкого набора типов нейронных сетей (генеративных, рекуррентных, сверточных и т. д.), что успешно доказано при апробации в предыдущем разделе.

Для оценки времени нахождения структур нейронных сетей и их итоговой точности при различных методах проведено несколько экспериментов:

- классификация объектов (датасет MNIST, функция ошибки — кросс-энтропия) [15];
- регрессия фрагмента данных на основе остальных значений (датасет Adult, функция ошибки — MSE) [16];
- прогнозирование временного ряда (датасеты Individual household electric power consumption, HAR accelerometer, функция ошибки — MSE) [16].

Получены следующие результаты (точность — в процентах, время поиска структуры и параметров нейронной сети — в секундах):

- классификация (MNIST): AutoKeras (99.8%, 2162), DEvol (99%, 3434), AutoSklearn (96.9%, 1897), предлагаемый метод (на основе автоэнкодера — 98.3%, 63, на основе рекуррентной сети — 98.1%, 192);
- регрессия (Adult): AutoKeras (86.9%, 954), DEvol (82.0%, 570), AutoSklearn (83.6%, 1121), предлагаемый метод (84.9%, 320);
- прогнозирование (Individual household electric power consumption): DEvol (99.7%, 2030), предлагаемый метод (99.9%, 1142);
- прогнозирование (HAR accelerometer): DEvol (94.8%, 1566), предлагаемый метод (92.9%, 868).

Не все методы использовались в каждом эксперименте, что обусловлено их ограничениями (см. таблицу). На рис. 4 и 5 представлены сводные диаграммы сравнения средней точности (в процентах) и среднего времени поиска структуры нейронной сети (в секундах) соответственно для каждого подхода.

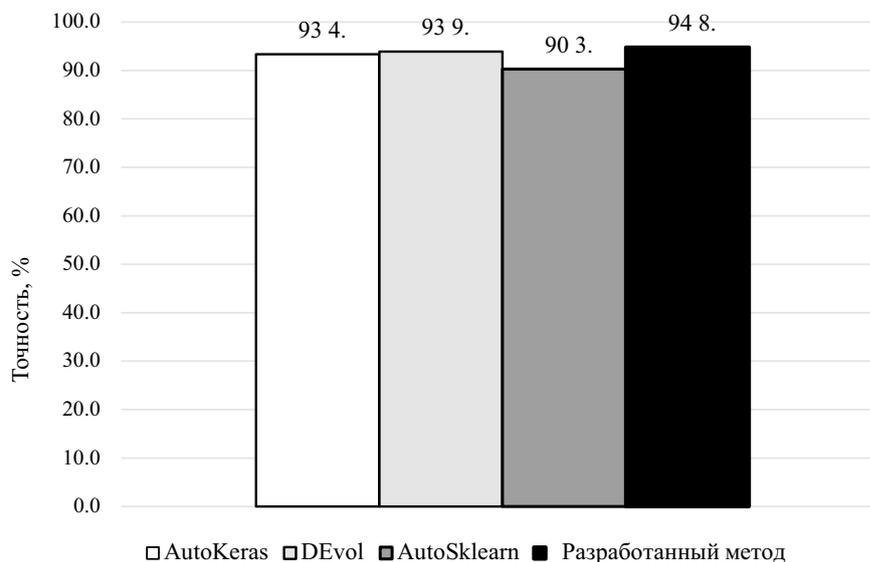


Рис. 4. Сравнение точности нейронных сетей, полученных различными методами поиска структуры нейронных сетей

Fig. 4. Comparison of the accuracy of neural networks obtained by various methods of searching for the structure of neural networks

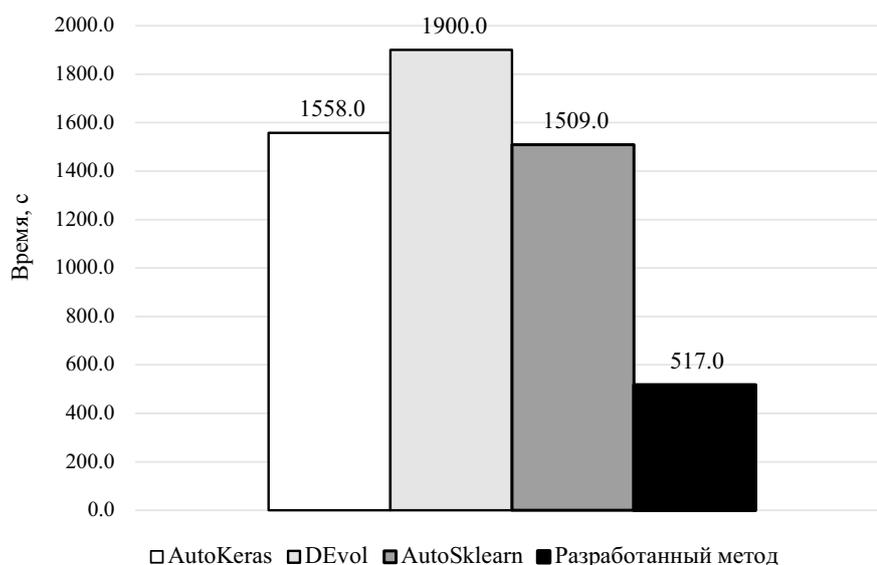


Рис. 5. Сравнение времени решения задачи поиска структуры нейронных сетей различными методами

Fig. 5. Comparison of the time of solving the problem of finding the structure of neural networks by different methods

Таким образом, средняя точность нейронных сетей после применения разработанного метода составила 94.8%, а среднее время поиска структуры и параметров сетей — 517 сек. Средняя точность остальных методов равна 92.5%. В среднем время решения задач остальными методами равнялось 1655 сек.

Полученные результаты сравнительного анализа показывают преимущество разработанного метода по времени нахождения оптимальных структур и параметров нейронных сетей произвольного типа, а также по итоговой точности полученных сетей.

## Заключение

Предложенный метод автоматического поиска структур и параметров нейронных сетей основан на определении категории задач обработки данных и переборе множеств варьируемых характеристик нейронных сетей на небольшой тестовой выборке, что способствует сокращению времени поиска при сохранении высокой точности итогового решения. Разработанный метод отличается от аналогов большей универсальностью, что позволяет использовать его для формирования многослойных, сверточных, рекуррентных, генеративно-состязательных нейронных сетей, а также автоэнкодеров.

Метод апробирован на примере двух задач (формирования автоэнкодера и генеративно-состязательной сети), относительно аналитического решения получено улучшение метрик на 26–34%. Сравнение метода при решении различных задач (классификация, регрессия, прогнозирование) с конкурентными программными библиотеками показало его превосходство по времени нахождения решения (до 3.2 раза) и итоговой точности нейронной сети (до 2.3%).

Таким образом, предложенный метод поиска структур и параметров достаточно прост в реализации, применим для широкого набора типов нейронных сетей и, соответственно, задач обработки информации, отличается быстроедействием и высокой



точностью. Область его применения включает разработку адаптивных информационных систем, интеллектуальных модулей анализа, обработки и генерации данных, систем, основанных на технологиях машинного зрения и обучения. В рамках проводимой в Тамбовском государственном техническом университете научной деятельности метод успешно апробирован при решении следующих задач: анализ данных о перемещении пользователя в виртуальной реальности, преобразование данных при передаче между компонентами тренажерных комплексов, персонализация интерфейса адаптивных систем электронного документооборота [17, 18].

### Список литературы

1. Бескровный А. С., Бессонов Л. В., Иванов Д. В., Кириллова И. В., Коссович Л. Ю. Использование сверточной нейронной сети для автоматизации построения двумерных твердотельных моделей позвонков // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2020. Т. 20, вып. 4. С. 502–516. <https://doi.org/10.18500/1816-9791-2020-20-4-502-516>
2. Li H., Yuan D., Ma X., Cui D., Cao L. Genetic algorithm for the optimization of features and neural networks in ECG signals classification // Scientific Reports. 2017. Vol. 7, iss. 1. P. 1–12. <https://doi.org/10.1038/srep41011>
3. He X., Zhao K., Chu X. AutoML: A Survey of the State-of-the-Art // Knowledge-Based Systems. 2021. Vol. 212. P. 106622. <https://doi.org/10.1016/j.knosys.2020.106622>
4. Jin H., Song Q., Hu X. Auto-keras: An efficient neural architecture search system // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019. P. 1946–1956. <https://doi.org/10.1145/3292500.3330648>
5. Real E., Aggarwal A., Huang Y., Le Q. V. Regularized evolution for image classifier architecture search // Proceedings of the AAAI Conference on Artificial Intelligence. 2019. Vol. 33. P. 4780–4789. <https://doi.org/10.1609/aaai.v33i01.33014780>
6. Budjac R., Nikmon M., Schreiber P., Zahradnikova B., Janacova D. Automated machine learning overview // Research Papers Faculty of Materials Science and Technology Slovak University of Technology. 2019. Vol. 27, iss. 45. P. 107–112. <https://doi.org/10.2478/rput-2019-0033>
7. Feurer M., Klein A., Eggenberger K., Springenberg J., Blum M., Hutter F. Efficient and robust automated machine learning // Automated Machine Learning. 2019. P. 113–134. [https://doi.org/10.1007/978-3-030-05318-5\\_6](https://doi.org/10.1007/978-3-030-05318-5_6)
8. Le Q., Zoph B. Using machine learning to explore neural network architecture // Google Research Blog. 2017. URL: <https://research.googleblog.com/2017/05/using-machine-learning-to-explore.html> (дата обращения: 05.04.2022).
9. Gong X., Chang S., Jiang Y., Wang Z. Autogan: Neural architecture search for generative adversarial networks // Proceedings of the IEEE International Conference on Computer Vision. 2019. P. 3224–3234. <https://doi.org/10.1109/ICCV.2019.00332>
10. Liu C., Zoph B., Neumann M., Shlens J., Hua W., Li L. J., Murphy K. Progressive Neural Architecture Search // Computer Vision – ECCV 2018. ECCV 2018 / eds.: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss. Cham : Springer, 2018. P. 19–34. (Lecture Notes in Computer Science, vol. 11205). [https://doi.org/10.1007/978-3-030-01246-5\\_2](https://doi.org/10.1007/978-3-030-01246-5_2)
11. Cai H., Gan C., Han S. Once for all: Train one network and specialize it for efficient deployment // arXiv preprint arXiv:1908.09791. 2019.
12. Truong A., Walters A., Goodsitt J., Hines K., Bruss C. B., Farivar R. Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools // 2019 IEEE 31st International Conference on Tools with Artificial Intelligence. 2019. P. 1471–1479. <https://doi.org/10.1109/ICTAI.2019.00209>
13. Moen E., Bannon D., Kudo T., Graf W., Covert M., Van Valen D. Deep learning for cellular image analysis // Nature Methods. 2019. Vol. 16, iss. 12. P. 1233–1246. <https://doi.org/10.1038/s41592-019-0403-1>



14. Obukhov A., Krasnyanskiy M. Quality assessment method for GAN based on modified metrics inception score and Frechet inception distance // Proceedings of the Computational Methods in Systems and Software. 2020. P. 102–114. [https://doi.org/10.1007/978-3-030-63322-6\\_8](https://doi.org/10.1007/978-3-030-63322-6_8)
15. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. URL: <http://yann.lecun.com/exdb/mnist/> (дата обращения: 05.04.2022).
16. UCI Machine Learning Repository. URL: <https://archive.ics.uci.edu/ml/datasets.php> (дата обращения: 05.04.2022).
17. Obukhov A., Siukhin A., Dedov D. The model of the automatic control system for a treadmill based on neural networks // 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). 2020. P. 1–5. <https://dx.doi.org/10.1109/FarEastCon50210.2020.9271589>
18. Обухов А. Д., Краснянский М. Н. Нейросетевой метод обработки и передачи данных в адаптивных информационных системах // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2021. Т. 31, вып. 1. С. 149–164. <https://doi.org/10.35634/vm210111>

### References

1. Beskrovny A. S., Bessonov L. V., Ivanov D. V., Kirillova I. V., Kossovich L. Yu. Using the mask-RCNN convolutional neural network to automate the construction of two-dimensional solid vertebral models. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2020, vol. 20, iss. 4, pp. 502–516 (in Russian). <https://doi.org/10.18500/1816-9791-2020-20-4-502-516>
2. Li H., Yuan D., Ma X., Cui D., Cao L. Genetic algorithm for the optimization of features and neural networks in ECG signals classification. *Scientific Reports*, 2017, vol. 7, iss. 1, pp. 1–12. <https://doi.org/10.1038/srep41011>
3. He X., Zhao K., Chu X. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 2021, vol. 212, pp. 106622. <https://doi.org/10.1016/j.knosys.2020.106622>
4. Jin H., Song Q., Hu X. Auto-keras: An efficient neural architecture search system. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1946–1956. <https://doi.org/10.1145/3292500.3330648>
5. Real E., Aggarwal A., Huang Y., Le Q. V. Regularized evolution for image classifier architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 4780–4789. <https://doi.org/10.1609/aaai.v33i01.33014780>
6. Budjac R., Nikmon M., Schreiber P., Zahradnikova B., Janacova D. Automated machine learning overview. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 2019, vol. 27, iss. 45, pp. 107–112. <https://doi.org/10.2478/rput-2019-0033>
7. Feuerer M., Klein A., Eggenesperger K., Springenberg J., Blum M., Hutter F. Efficient and robust automated machine learning. *Automated Machine Learning*, 2019, pp. 113–134. [https://doi.org/10.1007/978-3-030-05318-5\\_6](https://doi.org/10.1007/978-3-030-05318-5_6)
8. Le Q., Zoph B. Using machine learning to explore neural network architecture. *Google Research Blog*, 2017. Available at: <https://research.googleblog.com/2017/05/using-machine-learning-to-explore.html> (accessed 5 April 2022).
9. Gong X., Chang S., Jiang Y., Wang Z. Autogan: Neural architecture search for generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3224–3234. <https://doi.org/10.1109/ICCV.2019.00332>
10. Liu C., Zoph B., Neumann M., Shlens J., Hua W., Li L. J., Murphy K. Progressive Neural Architecture Search. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds.) *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, vol. 11205. Cham, Springer, 2018, pp. 19–34. [https://doi.org/10.1007/978-3-030-01246-5\\_2](https://doi.org/10.1007/978-3-030-01246-5_2)



11. Cai H., Gan C., Han S. Once for all: Train one network and specialize it for efficient deployment. *arXiv preprint*, arXiv:1908.09791. 2019.
12. Truong A., Walters A., Goodsitt J., Hines K., Bruss C. B., Farivar R. Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence*, 2019, pp. 1471–1479. <https://doi.org/10.1109/ICTAI.2019.00209>
13. Moen E., Bannon D., Kudo T., Graf W., Covert M., Van Valen D. Deep learning for cellular image analysis. *Nature Methods*, 2019, vol. 16, iss. 12, pp. 1233–1246. <https://doi.org/10.1038/s41592-019-0403-1>
14. Obukhov A., Krasnyanskiy M. Quality assessment method for GAN based on modified metrics inception score and Frechet inception distance. *Proceedings of the Computational Methods in Systems and Software*, 2020, pp. 102–114. [https://doi.org/10.1007/978-3-030-63322-6\\_8](https://doi.org/10.1007/978-3-030-63322-6_8)
15. *MNIST handwritten digit database*, Yann LeCun, Corinna Cortes and Chris Burges. Available at: <http://yann.lecun.com/exdb/mnist/> (accessed 5 April 2022).
16. *UCI Machine Learning Repository*. Available at: <https://archive.ics.uci.edu/ml/datasets.php> (accessed 5 April 2022).
17. Obukhov A., Siukhin A., Dedov D. The model of the automatic control system for a treadmill based on neural networks. *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2020, pp. 1–5. <https://dx.doi.org/10.1109/FarEastCon50210.2020.9271589>
18. Obukhov A., Krasnyanskiy M. Neural network method of data processing and transmission in adaptive information systems. *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2021, vol. 31, iss. 1, pp. 149–164 (in Russian). <https://doi.org/10.35634/vm210111>

Поступила в редакцию / Received 26.07.2021

Принята к публикации / Accepted 12.09.2022

Опубликована / Published 01.03.2023