



Применимость генетических алгоритмов для определения весовых коэффициентов искусственной нейронной сети с одним скрытым слоем

А. Д. Смородинов^{1,2*}, Т. В. Гавриленко^{1,2}, В. А. Галкин^{1,2}

¹ Сургутский филиал ФГУ ФНЦ НИИСИ РАН 628426, Тюменская область, Ханты-Мансийский автономный округ – Югра, г. Сургут, ул. Энергетиков, д. 4.

² БУ ВО «Сургутский государственный университет», 628412, Тюменская область, Ханты-Мансийский автономный округ – Югра, г. Сургут, пр. Ленина, д. 1

Аннотация. В обучении искусственной нейронной сети одной из центральных проблем является начальная инициализация и настройка весовых коэффициентов, связанная с псевдослучайной инициализацией весовых коэффициентов. В статье описан базовый генетический алгоритм, а также метод определения весовых коэффициентов с использованием данного алгоритма. Также представлен комбинированный метод определения весовых коэффициентов, предусматривающий на первом этапе начальную инициализацию с помощью генетического алгоритма и использование стохастического градиентного спуска на втором этапе обучения. Предлагаемые методы проверены на ряде искусственных нейронных сетей прямого распространения для различных задач по бинарной классификации реальных и синтетических данных, а также для однозначной многоклассовой классификации рукописных цифр на изображениях из базы данных MNIST. Искусственные нейронные сети сконструированы на основании теоремы Колмогорова-Арнольда. В данной статье проведен сравнительный анализ двух методов определения весовых коэффициентов – с использованием генетического алгоритма и градиентного спуска. По результатам сравнительного анализа сделан вывод о возможности применения генетического алгоритма для определения весовых коэффициентов как в качестве алгоритма начальной инициализации искусственной нейронной сети, так и в качестве алгоритма настройки весовых коэффициентов.

Ключевые слова: искусственные нейронные сети, генетический алгоритм, теорема Колмогорова-Арнольда, обучение нейронной сети.

Получение: 21.04.2023; Исправление: 08.06.2023; Принятие: 10.06.2023; Публикация онлайн: 29.06.2023

Для цитирования. Смородинов А. Д., Гавриленко Т. В., Галкин В. А. Применимость генетических алгоритмов для определения весовых коэффициентов искусственной нейронной сети с одним скрытым слоем // Вестник КРАУНЦ. Физ.-мат. науки. 2023. Т. 43. № 2. С. 69-86. EDN: YWFIZE. <https://doi.org/10.26117/2079-6641-2023-43-2-69-86>.

Финансирование. Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (Выполнение фундаментальных научных исследований ГП 47) по теме № 0580-2021-0007 «Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления».

Конкурирующие интересы. Конфликт интересов в отношении авторства и публикации нет.

Авторский вклад и ответственность. Авторы участвовали в написании статьи и полностью несут ответственность за предоставление окончательной версии статьи в печать.

*Корреспонденция: ✉ E-mail: Sachenka_1998@mail.ru

Контент публикуется на условиях Creative Commons Attribution 4.0 International License



© Смородинов А. Д., Гавриленко Т. В., Галкин В. А., 2023

© ИКИР ДВО РАН, 2023 (оригинал-макет, дизайн, составление)



Applicability of Genetic Algorithms for Determining the Weighting Coefficients of an Artificial Neural Network with One Hidden Layer

A. D. Smorodinov^{1,2}, T. V. Gavrilenko^{1,2}, V. A. Galkin^{1,2}*

¹ Surgut Branch of SRISA 628426, Surgut, Energetikov st., 4, Russia

² Surgut State University, 628412, Surgut, Lenina st., 1, Russia

Abstract. In the training of an artificial neural network, one of the central problems is the initial initialization and adjustment of weighting coefficients associated with pseudo-random initialization of weighting coefficients. The article describes a basic genetic algorithm, as well as a method for determining weight coefficients using this algorithm. A combined method for determining weighting coefficients is also presented, which provides for initial initialization using a genetic algorithm at the first stage and the use of stochastic gradient descent at the second stage of training, the proposed methods are tested on a number of artificial neural networks of direct propagation for various tasks of binary classification of real and synthetic data, as well as for unambiguous multiclass classification of handwritten digits on images from the database MNIST data. Artificial neural networks are constructed on the basis of the Kolmogorov-Arnold theorem. This article presents a comparative analysis of two methods for determining weight coefficients – using a genetic algorithm and gradient descent. Based on the results of the comparative analysis, it is concluded that a genetic algorithm can be used to determine the weighting coefficients both as an algorithm for the initial initialization of an artificial neural network and as an algorithm for adjusting the weighting coefficients.

Key words: artificial neural networks, genetic algorithm, Kolmogorov-Arnold theorem, neural network training.


Received: 21.04.2023; Revised: 08.06.2023; Accepted: 10.06.2023; First online: 29.06.2023

For citation. Smorodinov A. D., Gavrilenko T. V., Galkin V. A. Applicability of Genetic Algorithms for Determining the Weighting Coefficients of an Artificial Neural Network with One Hidden Layer. *Vestnik KRAUNC. Fiz.-mat. nauki.* 2023, 43: 2, 69-86. EDN: YWFIZE. <https://doi.org/10.26117/2079-6641-2023-43-2-69-86>.

Funding. The publication was made within the framework of the state task of the Federal State Institution FNTs NIISI RAS (Performance of fundamental scientific research GP 47) on topic No. 0580-2021-0007 «Development of methods for mathematical modeling of distributed systems and corresponding calculation methods» here

Competing interests. There are no conflicts of interest regarding authorship and publication.

Contribution and Responsibility. All authors contributed to this article. Authors are solely responsible for providing the final version of the article in print. The final version of the manuscript was approved by all authors.

*Correspondence:  E-mail: Sachenka_1998@mail.ru

The content is published under the terms of the Creative Commons Attribution 4.0 International License

© Smorodinov A. D., Gavrilenko T. V., Galkin V. A., 2023

© Institute of Cosmophysical Research and Radio Wave Propagation, 2023 (original layout, design, compilation)



Введение

Ключевой проблемой искусственных нейронных сетей является инициализация весовых коэффициентов, а также сходимость сети для решения задач. Данная проблема обсуждалась в [1], авторы показывали, что сеть одной конфигурации при многократных обучении может показывать разные результаты: одни показывали низкий уровень сходимости, другие показывали достаточно хорошие результаты с погрешностью в $10e-1$, а некоторые показывали высокую сходимость на уровне $10e-5$. Такая ситуация складывается в результате того, что весовые коэффициенты нейронной сети инициализировались псевдослучайным образом, что приводит к тому, что качество сходимости нейронной сети зависит от алгоритма инициализации весовых коэффициентов. Генетический алгоритм есть эвристический алгоритм поиска, который впервые был описан в работе [2] и изначально был назван «репродуктивным планом Холланда», по фамилии его создателя. Автор применял его в качестве основного алгоритма в различных эволюционных алгоритмах. В дальнейшем алгоритм был доработан, и уже в работе [3] был создан алгоритм, который сейчас принято называть классическим генетическим алгоритмом.

В настоящее время генетический алгоритм используется в различных областях науки, так, например в [5] авторы применяют генетический алгоритм для выбора структуры информационно-вычислительных систем. В работе [6] авторы используют генетический алгоритм для восстановления траектории движения воздушного судна. Данный алгоритм нашел применение и в алгоритмах шифрования, так, в работах [7, 8] авторы применяют генетический алгоритм для разработки симметричного алгоритма шифрования. Генетический алгоритм также используется для обучения искусственных нейронных сетей, к примеру, в работе [9] автор предлагает использовать его для того, чтобы определить, какие именно весовые коэффициенты в искусственной нейронной сети необходимы, а от каких можно отказаться. Однако автор ограничился только теоретическим описанием алгоритма, не проводя никаких вычислительных экспериментов, показывающих работоспособность предложенного им алгоритма.

В данной статье представлены результаты определения значений весовых коэффициентов искусственной нейронной сети на основе генетического алгоритма.

Базовый генетический алгоритм

Генетический алгоритм направлен на поиск глобальных экстремумов функции. Пусть требуется найти глобальный минимум некоторой функции

$$y = f(x_1, x_2, \dots, x_n),$$

т.е. необходимо подобрать такие значения x_1, x_2, \dots, x_n , при которых значение y будет минимально. Для этого создаются несколько множеств, или как принято называть в генетических алгоритмах популяция $A^i = \{x_1^i, \dots, x_n^i\}$, где $i = 1, \dots, m, m$

- число множеств, или, как их ещё называют, число особей или индивидов в популяции.

Далее задается функция приспособленности, которая заключается в том, что рассчитываются

$$y = f(A^i)$$

и выбирается k особей, для которых данная функция имеет минимальное значение. После чего применяется операция кроссинговера или скрещивания, которая заключается в перестановке случайной переменной x_1, x_2, \dots, x_n или сразу нескольких с этими же значениями другой особи. Следующим шагом проводят с некоторой вероятностью операцию мутации, случайное изменение значения x_1, x_2, \dots, x_n в случайной особи. После чего заново рассчитывают значение функции приспособленности и проверяют, выполнены ли условия останова алгоритма. Если нет, то проводим отбор «лучших особей», для которых функция принимает наименьшее значение и т.д. Блок-схема классического генетического алгоритма представлена на рис.1.

Искусственная нейронная сеть

Искусственную нейронную сеть можно записать в следующем виде:

$$f(x_1, x_2) = G'''(w_1^{2,1}G''(w_1^{1,1}G'(x_1) + w_1^{1,2}G'(x_2))) + w_1^{2,2}G''(w_2^{1,1}G'(x_1) + w_2^{1,2}G'(x_2)) + w_1^{2,3}G''(w_3^{1,1}G'(x_1) + w_3^{1,2}G'(x_2)) \quad (1)$$

Данная формула описывает двухслойную ИНС с двумя входами и одним выходом. Задача обучения ИНС сводится к подбору весовых коэффициентов w , обычно для этих целей используют алгоритмы градиентного спуска. В данной статье для подбора весовых коэффициентов предлагается использовать генетический алгоритм, где в качестве особи генетического алгоритма используется массив из всех весовых коэффициентов.

Алгоритм, используемый для подбора весовых коэффициентов, отличается от классического большим количеством генов в индивиде, а, соответственно, и большей вычислительной сложностью, чем применение классического генетического алгоритма.

Вначале генерировалось 40 особей и проверялось, какая из особей имела минимальное значение функции потерь, далее над этой особью проводились определенные действия. Но перед этим обозначим особь за X , множество X разделим пополам и обозначим первую половину как A , а вторую половину как B . В произвольном месте множеств A и B изменим значение x на некоторую случайную величину, измененные множества A и B обозначим как A', B' соответственно и составим следующие четыре множества, или популяцию, состоящую из 4 особей:

$$X^1 = \{A, B\} = X \text{ - т.е. исходное множество. } X^2 = \{A', B\}, X^3 = \{A, B'\}, X^4 = \{A', B'\}.$$



Рис. 1. Блок-схема работы генетического алгоритма
 [Figure 1. Block diagram of the genetic algorithm]

После этого проверяется, какое из множеств лучше удовлетворяет условию задачи, остальные множества удаляются, после чего описанный алгоритм повторяется до тех пор, пока не будет найдено такое множество, которое полностью

удовлетворяет условию задачи, либо пока время, отведенное на работу алгоритма, не закончится. Время работы алгоритма измеряется в итерациях.

Вычислительные эксперименты проводились с использованием следующих аппаратно-программных средств. Аппаратная часть состояла из процессора AMD Ryzen Threadripper 2990WX 32-Core Processor, 32 ГБ оперативной памяти с частотой 2400 МГц, графический процессор Nvidia Titan V. Программная часть была реализована на языке программирования Python с использованием библиотек Keras и TensorFlow. Реализация генетического алгоритма была также написана на языке Python без использования сторонних библиотек.

Далее будут представлены результаты вычислительных экспериментов по определению весовых коэффициентов ИНС с помощью генетического алгоритма на различных задачах.

Вычислительные эксперименты

В данной статье в первой части вычислительных экспериментов рассматривается задача бинарной классификации. Согласно [4] на первом слое использовалась функция активации Relu (2), Relu – линейный выпрямитель с утечкой [10, 11].

$$\text{Relu}(x) = \begin{cases} ax & x < 0 \\ x & x \geq 0 \end{cases} \quad (2)$$

a принимает значение $[1, 0]$, если $a = 0$, то данная функция является линейным выпрямителем без утечки.

На выходном слое Sigmoid (3) – это гладкая монотонная возрастающая функция, нелинейная. Часто её используют для «сглаживания» некоторой величины, в нейронных сетях данную функцию активации так же, как и гиперболический тангенс, используют для решения задачи регрессии между 0 и 1, применяя на выходном слое нейронной сети.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

В качестве алгоритма оптимизации использовался RMSprop. RMSprop является модификацией алгоритма adagrad и заключается в том, что для обновления весов используется не полная сумма обновлений, а усреднённый по истории квадрат градиента [12].

В качестве функции потерь выбрана функция Binarycrossentropy (4). Binarycrossentropy (BCE) – бинарная перекрестная энтропия. Формула данной функции:

$$\sum_{i=1}^n (\max(y_i, 0) - y_i * x_i + \log(1 + \exp(-|y_i|))) \quad (4)$$

Согласно теореме Колмогорова о существовании отображающей нейронной сети.

Теорема (теорема Колмогорова о существовании отображающей нейронной сети). Для любой непрерывной функции $f: I^n \rightarrow R^m, f(x) = y$, где I

– замкнутый единичный интервал $[0, 1]$ (следовательно, Γ^n – мерный единичный куб), f может быть реализована с требуемой точностью трехслойной нейронной сетью, имеющей n нейронов $x = (x_1, \dots, x_n)$ в первом слое, $2n+1$ в скрытом слое и u нейронов в выходном слое.

Доказательство теоремы представлено в работе [13].

Согласно этой теореме ИНС будет содержать 1 скрытый слой с функцией активации, которая была выбрана экспериментальным путем.

В рамках решаемой задачи лучшей показала себя функция (5) SELU. SELU –масштабированная экспоненциальная линейная функция (Scaledexponentiallinearunit, SELU) [14, 15].

$$SELU(a, x) = \delta \begin{cases} a(e^x - 1)x < 0 \\ xx \geq 0 \end{cases} \quad (5)$$

где $\delta = 1,0507$, $a = 1,67326$.

По факту $SELU(x) = \delta * ELU(x, a)$, где δ , a являются преопределенными константами. Данные значения выбирают так, чтобы среднее и дисперсия входов сохранялись между двумя последовательными слоями до тех пор, пока веса правильно инициализированы, а количество входов достаточно велико.

Вычислительный эксперимент №1

Для первого вычислительного эксперимента была выбрана задача классификации 60 женщин на две категории на основе 5 признаков. Часть обучающей выборки представлена в табл. 1

Таблица 1

Исходные данные для ИНС.
[Initial data for artificial neural networks]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5	Выход 1	Выход 2
1	6	6	83	35	71	1	0
2	3	15	98	50	43	1	0
3	11	4	114	25	174	1	0
4	3	14	92	46	53	1	0
5	4	13	89	42	57	1	0
6	3	15	67	58	24	0	1
7	4	10	91	44	53	0	1
8	4	17	88	69	27	0	1
9	1	16	81	67	19	0	1
10	2	17	78	60	23	0	1

Нейронную сеть обучали в течение 50 эпох, в результате на 50 эпохе потери НС составили 0.6466. Результаты представлены в табл.2

Таблица 2

Применение обученной ИНС для классификации женщин на две категории без участия генетического алгоритма.
[The use of a trained artificial neural networks to classify women into two categories without the participation of a genetic algorithm]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	1	18	76	74	15
2	2	16	86	53	34
3	7	8	105	46	95
4	3	8	83	39	46
5	7	8	98	29	108
6	6	7	89	33	75
7	6	9	86	39	70
8	2	15	81	62	25
9	8	6	93	26	132
10	2	17	78	60	23
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,42656043	0,5976147	0	1	+
2	0,42656043	0,5976147	0	1	+
3	0,4553618	0,59136885	1	0	-
4	0,39758042	0,43312028	1	0	-
5	0,8907546	0,0751419	1	0	+
6	0,46376246	0,60819185	1	0	-
7	0,4163843	0,4985347	0	1	+
8	0,42656043	0,5976147	0	1	+
9	0,9557723	0,05102192	1	0	+
10	0,42656043	0,5976147	0	1	+

Выходы НС необходимо интерпретировать как вероятность того, что набор данных относится к классу, равному номеру выхода, так, например, в 9 наборе данных ИНС показывает, что с вероятностью в 0.97 этот набор данных относится к 1 классу, что соответствует ожиданиям. В итоге ИНС в 37 случаях из 60 правильно классифицировала их.

Далее для данной ИНС и вышеописанной задачи применялся, применяя генетический алгоритм.

В качестве функции приспособленности будем применять функцию потерь Binarycrossentropy, которую использовали для оценки качества обучения ИНС. Качество работы ИНС с подбором весовых коэффициентов с использованием генетического алгоритма.

При создании нейронной сети потери составляли 0.973, но уже к 60 итерации потери составили 0.669, к 300 итерации алгоритма – 0.654, а при максимально возможных 10 000 итераций потери достигли 0.6146. В табл.3 представлен результат работы ИНС после подбора весовых коэффициентов с помощью генетического алгоритма.

Таблица 3

Результаты применения ИНС, в которой весовые коэффициенты определялись с помощью генетического алгоритма.
 [The results of the application of the artificial neural networks, in which the weight coefficients were determined using a genetic algorithm]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	1	18	76	74	15
2	2	16	86	53	34
3	7	8	105	46	95
4	3	8	83	39	46
5	7	8	98	29	108
6	6	7	89	33	75
7	6	9	86	39	70
8	2	15	81	62	25
9	8	6	93	26	132
10	2	17	78	60	23
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,47827497	0,52794385	0	1	+
2	0,47827497	0,52794385	0	1	+
3	0,3867269	0,6394901	1	0	-
4	0,47827497	0,52794385	1	0	-
5	0,5635742	0,19807903	1	0	+
6	0,4070327	0,65066314	1	0	-
7	0,36243546	0,627527	0	1	+
8	0,47827497	0,52794385	0	1	+
9	0,9761327	0,05350576	1	0	+
10	0,47827497	0,52794385	0	1	+

В результате анализа применения ИНС было определено, что в 38 из 60 случаев сеть правильно классифицировала данные.

После подбора коэффициентов обучаем ИНС в течение 50 эпох с использованием RMSprop, значение оценки потери стало равным 0.6166, т.е. после обучения ИНС стала хуже классифицировать данные, результаты применения ИНС представлены в табл. 4

Исходя из анализа результатов определено, что ИНС только в 37 из 60 случаев правильно классифицировала данные. Исходя из вышесказанного, можно сделать вывод, что дообучения ИНС, для которой были подобраны весовые коэффициенты, не требуется.

Возможно, проблема заключается в достаточно сложной задаче классификации ввиду малого количества исходных данных, поэтому в следующем эксперименте решалась задача классификации синтетических данных.

Таблица 4

Результаты применения ИНС, в которой весовые коэффициенты определялись комбинированным способом.

[The results of the application of the artificial neural networks, in which the weight coefficients were determined by a combined method]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	1	18	76	74	15
2	2	16	86	53	34
3	7	8	105	46	95
4	3	8	83	39	46
5	7	8	98	29	108
6	6	7	89	33	75
7	6	9	86	39	70
8	2	15	81	62	25
9	8	6	93	26	132
10	2	17	78	60	23
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,4559949	0,5416433	0	1	+
2	0,4559949	0,5416433	0	1	+
3	0,32548302	0,67317367	1	0	-
4	0,4559949	0,5416433	1	0	-
5	0,86289877	0,11995316	1	0	+
6	0,34544602	0,66923875	1	0	-
7	0,2917664	0,6884599	0	1	+
8	0,4559949	0,5416433	0	1	+
9	0,99906164	0,00234613	1	0	+
10	0,4559949	0,5416433	0	1	+

Вычислительный эксперимент №2

В качестве генератора исходных данных было взято выражение следующего вида:

$$F(x_1, x_2, x_3, x_4, x_5) = 3 * x_1 + x_2 - x_3 + 4 * x_4 - x_5^2 \quad (6)$$

и произвольный набор x . Если выражение 6 принимало значение большее либо равное 0, то это классифицировалось как 0, 1, в противном случае – как 1, 0. Часть обучающей выборки представлена в Табл.5. Всего было сгенерировано 60 наборов, по 30 для каждой из групп.

Применяемая ИНС и алгоритм обучения были аналогичны эксперименту 1, на 50 эпохе потери составляли 0.2327, в табл. 6 представлены результаты применения обученной нейронной сети.

По результатам анализа обучения видно, что в 55 из 60 случаев ИНС правильно классифицировала данные. Во второй части эксперимента весовые коэффициенты для ИНС определялись с использованием генетического алгоритма.

Созданная нейронная сеть до подбора коэффициентов имела потери, равные 3.24, к 50 итерации потери составляли 0.889, к 100 – 0.725, к 1000 – 0.12, а к 10 000

Таблица 5

**Обучающая выборка на основе формулы (6).
[Training sample based on formula (6)]**

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5	Выход 1	Выход 2
1	50	89	-22	-28	3	1	0
2	31	-75	12	37	2	1	0
3	-13	-18	-65	55	-7	1	0
4	-3	31	33	73	-5	1	0
5	-67	82	0	42	-2	1	0
6	-48	82	-93	-53	-26	0	1
7	-40	-45	83	-97	-74	0	1
8	-59	46	-40	-61	50	0	1
9	9	78	32	-28	-64	0	1
10	47	10	-93	85	30	0	1

итерации равнялись 0.05738. Результаты применения ИНС с весами, подобранными с помощью генетического алгоритма, представлены в табл. 7

Как видно из таблицы 7, результат подбора коэффициентов неудовлетворительный. ИНС правильно классифицирует данные только в 7 случаях из 10. После анализа всего набора данных было выявлено, что в 11 из 60 случаев ИНС некорректно классифицировала наборы данных.

После ИНС была обучена аналогично эксперименту 1. На 50 эпохе потери составляли 0.03, что больше, чем в эксперименте 1. Результаты применения обученной ИНС представлены в табл.8

По итогам анализа результатов определено, что только в 1 случае из всей выборки объемом в 60 наборов данных ИНС не смогла правильно классифицировать данные. Отсюда можно сделать вывод, что после применения генетического алгоритма для подбора весовых коэффициентов необходимо дообучить ИНС с помощью классических методов.

Вычислительный эксперимент №3

Две вышеописанные задачи являются все-таки относительно простыми, и потому в следующем эксперименте ИНС обучали для классификации изображений. Изображения представляют собой рукописные черно-белые цифры размером 28x28 пикселей, разбитые на 10 категорий (от 0 до 9). Использовался стандартный набор из 60 000 изображений, собранных Национальным институтом стандартов и технологий США в 1980-х годах, который носит название MNIST. Образцы изображений из набора MNIST представлены на рисунке 2

Для данной задачи были внесены изменения в размер ИНС, но правила конструирования остались неизменными. С учетом размерности входных данных размер нейронной сети составил 784 нейрона на входном слое, 1569

Таблица 6

Применение обученной ИНС без использования генетического алгоритма.
[Application of a trained artificial neural networks without using a genetic algorithm]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	97	-84	-89	20	40
2	48	41	94	91	-77
3	97	-73	-27	84	9
4	16	20	31	89	6
5	49	27	42	99	1
6	-1	64	-61	61	2
7	77	-99	14	61	-69
8	-70	63	-97	-91	-8
9	-67	63	-53	40	4
10	-80	-72	64	35	-54
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,83027977	0,16956526	0	1	-
2	6,9131456e-05	9,9763930e-01	0	1	+
3	0,83027977	0,16956526	1	0	+
4	0,9050562	0,101865	1	0	+
5	0,83027977	0,16956526	1	0	+
6	0,83027977	0,16956526	1	0	+
7	0,83027977	0,16956526	0	1	-
8	3,707162e-11	1,000000e+00	0	1	+
9	9,9986172e-01	6,9025705e-06	1	0	+
10	5,397542e-11	1,000000e+00 0	0	1	+

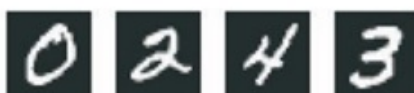


Рис. 2. Образцы изображений MNIST
 [Figure 2. MNIST Image Samples]

нейронов на скрытом слое и 10 нейронов на выходном слое. Остальные характеристики остались без изменений

Вообще говоря, задачи по распознаванию изображений стоит решать с использованием сверточных нейронных сетей, но в данном случае и классическая ИНС справится с поставленной задачей [10].

Как уже было сказано, данная задача решена, однако возникает вопрос, можно ли улучшить работу ИНС, если подбор весовых коэффициентов будет проходить с использованием генетического алгоритма, описанного выше.

В данном эксперименте подбор весовых коэффициентов происходил с использованием генетического алгоритма. На 0 итерации нейронная сеть имела

Таблица 7

Результаты применения ИНС, в которой весовые коэффициенты определялись с помощью генетического алгоритма.

[The results of the application of the artificial neural networks, in which the weight coefficients were determined using a genetic algorithm]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	97	-84	-89	20	40
2	48	41	94	91	-77
3	97	-73	-27	84	9
4	16	20	31	89	6
5	49	27	42	99	1
6	-1	64	-61	61	2
7	77	-99	14	61	-69
8	-70	63	-97	-91	-8
9	-67	63	-53	40	4
10	-80	-72	64	35	-54
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,9405218	0,989976	0	1	+
2	0,96083605	0,93867195	0	1	-
3	9,9943906e-01	3,8614900e-05	1	0	+
4	9,999093e-01	9,721195e-05	1	0	+
5	1	0,00490132	1	0	+
6	0,5091303	0,55081743	1	0	-
7	0,5091303	0,55081743	0	1	+
8	1,3349686e-08	1,0000000e+00	0	1	+
9	0,99533975	0,00313419	1	0	+
10	2,360904e-05	1,000000e+00	0	1	+

потери, равные 0.3246, к 500 итерации погрешность составила 0.3029. Данный шаг был крайним в этом эксперименте ввиду большого размера нейронной сети. После проверки выяснилось, что ИНС только в 18 135 случаях правильно классифицировала рукописные изображения.

Далее нейронная сеть была обучена по стандартному алгоритму обучения ИНС с использованием модифицированного стохастического градиентного спуска, обучение проходило в течение 10 эпох, и потери составили 0.00498. После проверки ИНС выявлено, что только в 330 случаях из 60 000 нейронная сеть неверно классифицировала данные, что является достаточно хорошим результатом, если учесть, что человек не всегда способен классифицировать рукописные символы на 100 %.

Следующим этапом проверялось, есть ли эффект от генетического алгоритма или в данном случае ИНС стоит сразу обучать с использованием алгоритма RMSprop.

Для этого была взята та же самая ИНС, с теми же начальными весовыми коэффициентами (до подбора весовых коэффициентов с использованием генетического алгоритма) и обучена в течение 10 эпох. Потери составили 0.00435,

Таблица 8

Результаты применения ИНС, в которой весовые коэффициенты определялись комбинированным способом. [The results of the application of the artificial neural networks, in which the weight coefficients were determined by a combined method]

№	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5
1	97	-84	-89	20	40
2	48	41	94	91	-77
3	97	-73	-27	84	9
4	16	20	31	89	6
5	49	27	42	99	1
6	-1	64	-61	61	2
7	77	-99	14	61	-69
8	-70	63	-97	-91	-8
9	-67	63	-53	40	4
10	-80	-72	64	35	-54
11	11	88	-84	84	-28
№	Выход НС 1	Выход НС 2	Ожидаемый выход 1	Ожидаемый выход 2	Оценка НС
1	0,01893292	0,8971423	0	1	+
2	6,2430176e-05	9,9997222e-01	0	1	+
3	9,9999535e-01	6,7443682e-09	1	0	+
4	9,9961156e-01	5,2609116e-06	1	0	+
5	9,9999034e-01	4,5175893e-06	1	0	+
6	0,57582635	0,47869515	1	0	+
7	6,0377239e-05	9,9999774e-01	0	1	+
8	2,0454194e-15	1,0000000e+00	0	1	+
9	9,9994063e-01	2,4438985e-05	1	0	+
10	4,7350012e-36	1,0000000e+00	0	1	+
11	0,57771194	0,47786874	0	1	-

что немного лучше, чем в прошлый раз. После проверки всей обучающей выборки с использованием данной ИНС получено, что только в 187 случаях ИНС неверно классифицировала изображение, что также является лучшим результатом. В результате получается, что ИНС, обученная без использования генетического алгоритма, на 1.89 % качественнее обучена по сравнению с ИНС, обученной с использованием генетического алгоритма.

Данную разницу можно отнести к погрешности, таким образом, можно сказать, что нейронные сети обучены одинаково. Если посмотреть на количество весовых коэффициентов в ИНС для данной задачи и для задачи из вычислительного эксперимента 1, то видно, что первом случае весовых коэффициентов $5 \cdot 10 + 10 \cdot 2 = 70$, а во втором случае $- 784 \cdot 1569 + 1569 \cdot 10 = 1\ 245\ 786$, т.е. разница в размере особи более чем в 17 000. На основе этого можно предположить, что генетический алгоритм показал слабые результаты из-за того, что в прошлых вычислительных экспериментах мутация приходилась на 1 из 35 генов ИНС против 1 на 622 000 генов в данном эксперименте. Учитывая этот факт, был

модифицирован генетический алгоритм и увеличилось количество мутаций на особь.

В модифицированном варианте генетического алгоритма множество разделяется на 4 равных части, и уже среди 5 частей происходит комбинирование для создания новых особей, остальное в данном алгоритме остается без изменений.

Подбор весовых коэффициентов с использованием модифицированного генетического алгоритма проходил в течение 500 итераций, в итоге потери составили 0.2859, что является лучшим результатом по сравнению с генетическим алгоритмом, используемым предыдущем эксперименте. После проверки применения ИНС для классификации изображений получено, что в 26 833 случаях ИНС правильно определяла цифру. Это показывает, что модифицированный алгоритм работает лучше. Далее ИНС дообучили с использованием стандартных методов в течение 10 эпох. После обучения потери составили 0.0045 (в предыдущем эксперименте значение потерь составляла 0.00498), и только в 471 случае ИНС неверно классифицировала данные.

Заключение

В данной статье рассмотрена связь задачи обучения искусственных нейронных сетей с генетическим алгоритмом, показано, что обучение ИНС можно представить в виде одного из видов генетического алгоритма. Проведена серия вычислительных экспериментов, в которых показано, что применение генетического алгоритма для подбора весовых коэффициентов ИНС с одним скрытым слоем не лишено смысла и в определенных случаях действительно может показать результат лучше, чем обучение с использованием модифицированного стохастического градиентного спуска. Тем не менее вопрос о применении данного подхода к подбору коэффициентов ИНС большего размера остается открытым, возможно, для подбора коэффициентов в этом случае требуется использование не классического генетического алгоритма. -


Список литературы

1. Галкин В. А., Гавриленко Т. В., Смородинов А. Д. Некоторые аспекты аппроксимации и интерполяции функций искусственными нейронными сетями, *Вестник КРАУНЦ. Физ.-мат. науки*, 2022. Т. 38, № 1, С. 54-73 DOI: 10.26117/2079-6641-2022-38-1-54-73.
2. Holland J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge: The MIT Press, 1992.
3. Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Massachusetts: Addison-Wesley, 1989.
4. Шолле Ф. Глубокое обучение на Python, 2018. 400 с.
5. Zakharov I. V. , Shushakov A. O., Zykova S. S. The Choice of Structures of Heterogeneous Information-Computer Systems Based on the Apparatus of Genetic Algorithms, *Интеллектуальные технологии на транспорте*, 2022. № 3(31), С. 46-51 DOI: 10.24412/2413-2527-2022-331-46-51.
6. Бедин Д. А., Иванов А. Г. Использование генетического алгоритма для определения параметров многогипотезного алгоритма восстановления траектории воздушного судна / *XXVI Санкт-Петербургская международная конференция по интегрированным навигационным системам*, Сборник материалов, Санкт-Петербург, 27–29 мая 2019 года. Санкт-Петербург, "Концерн "Центральный научно-исследовательский институт "Электроприбор", 2019, С. 87-90.


7. Алхуссайн А. Х. Симметричный алгоритм шифрования с помощью генетического алгоритма и генераторов псевдослучайных чисел, *Естественные и технические науки*, 2015. № 7(85), С. 75-81.
8. Стефанюк В. Л., Алхуссайн А. Х. Криптография с симметричным ключом с использованием генетического алгоритма / *КИИ-2014*, Четырнадцатая национальная конференция по искусственному интеллекту с международным участием, Т. 1. Казань, РИЦ «Школа», 2014, С. 267-275.
9. Трокоз Д. А. Алгоритм машинного обучения широких нейронных сетей с использованием алгебры гиперразмерных двоичных векторов и генетических алгоритмов, *Южно-Сибирский научный вестник*, 2020. № 6(34), С. 148-154.
10. Nair V., Hinton G.E. Rectified Linear Units Improve Restricted Boltzmann Machines / *27th International Conference on International Conference on Machine Learning*. USA, Omnipress, 2010, pp. 807–814.
11. Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models, *Proc. ICML*, 2013. vol. 30, no. 1.
12. Tieleman, Tijmen and Hinton, Geoffrey *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural Networks for Machine Learning, 2012.
13. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения, *Докл. АН СССР*, 1957. Т. 114, № 5, С. 953–956.
14. Klambauer G., Unterthiner T., Mayr A., Hochreiter S. Self-Normalizing Neural Networks, *Advances in Neural Information Processing Systems*, 2017. vol. 30, no. 2017, arXiv: 170602515k.
15. Clevert, Djork-Arné; Unterthiner, Thomas Hochreiter *Fast and Accurate Deep Network Learning by Exponential Linear Units*, 2015, arXiv: 1511.07289.

Информация об авторах




Смородинов Александр Денисович ✉ – БУ ВО «Сургутский государственный университет», Аспирант кафедры прикладной математики, преподаватель кафедры АСОИУ, Сургутский филиал ФГУ «ФНЦ НИИСИ РАН» Инженер отдела биофизики,  ORCID 0000-0002-9324-1844.



Гавриленко Тарас Владимирович ✉ – БУ ВО «Сургутский государственный университет», доцент Сургутский филиал ФГУ «ФНЦ НИИСИ РАН» Заместитель директора.  ORCID 0000-0002-3243-2751.



Галкин Валерий Алексеевич ✉ – БУ ВО «Сургутский государственный университет», профессор, доктор физико-математических наук. Сургутский филиал ФГУ «ФНЦ НИИСИ РАН» Директор.  ORCID 0000-0002-9721-4026.

References

- [1] Galkin V. A., Gavrilenko T. V., Smorodinov A. D. Some aspects of approximation and interpolation of functions artificial neural networks. Vestnik KRAUNC. Fiz.-mat. nauki. 2022, 38: 1, 54-73. DOI: 10.26117/2079-6641-2022-38-1-54-73, (In Russian).
- [2] Holland J. H. Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Cambridge, 1992.
- [3] Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. Massachusetts: Addison-Wesley, 1989.
- [4] Chollet F. Deep Learning with Python. Simon and Schuster, N.Y., USA. 2021, p. 384.
- [5] Zakharov I. V. , Shushakov A. O., Zykova S. S. The Choice of Structures of Heterogeneous Information-Computer Systems Based on the Apparatus of Genetic Algorithms. Transportda intellektual texnologiyalar , 2022, 3(31), 46-51, DOI: 10.24412/2413-2527-2022-331-46-51
- [6] Bedin D. A., Ivanov A. G. Ispol'zovaniye geneticheskogo algoritma dlya opredeleniya parametrov mnogogipoteznogo algoritma vosstanovleniya trayektorii vozdušnogo sudna. XXVI Sankt-Peterburgskaya mezhdunarodnaya konferentsiya po integrirovannym navigatsionnym sistemam. Sbornik materialov, Sankt-Peterburg, 27–29 maya 2019 goda, Sankt-Peterburg, "Kontsern "Tsentral'nyy nauchno-issledovatel'skiy institut "Elektropribor 2019, 87-90, (In Russian).
- [7] Alhussain A. H. Symmetric encryption algorithm using genetic algorithm and pseudo-random number generators. Yestestvennyye i tekhnicheskiye nauki, 2015, 7(85), 75-81 (In Russian).
- [8] Stefanyuk V. L., Alkhussayn A. KH. Kriptografiya s simmetrichnym klyuchom s ispol'zovaniyem geneticheskogo algoritma, KII-2014. Chetyrnadtsataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiyem. T. 1, Kazan': RITS «Shkola», 2014, 267-275 (In Russian).
- [9] Trokoz D. A. Machine learning algorithm for wide neural networks using hyperdimensional binary vector algebra and genetic algorithms. Yuzhno-Sibirskiy nauchnyy vestnik, 2020, 6(34), 148-154 (In Russian).
- [10] Nair V., Hinton G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. 27th International Conference on International Conference on Machine Learning. USA: Omnipress, 2010, 807–814.
- [11] Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models, Proc. ICML. 2013, 30, 1.
- [12] Tieleman T., Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning. 2012.
- [13] Kolmogorov A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, Doklady Akademii Nauk SSSR, 1957, 114, 5, P. 953–956, (In Russian).
- [14] Klambauer G., Unterthiner T., Mayr A., Hochreiter S. Self-Normalizing Neural Networks, Advances in Neural Information Processing Systems. 2017, 30, 2017.
- [15] Clover Y.-A., Untertiner T., Hochreiter S. Fast and accurate deep network learning using exponential linear units, arXiv preprint, arXiv:1511.07289, 2015.

Information about authors



Smorodinov Aleksandr Denisovich ✉ – Surgut State University, Postgraduate Student of the Department of Applied Mathematics, Lecturer of the Department of ASOIU, Branch of SRISA, Engineer of the Department of Biophysics and Neurocybernetics.
ORCID 0000-0002-9324-1844.



Gavrilenko Taras Vladimirovich ✉ – Ph.D. (Tech.), Assistant Professor, Surgut State University, Branch of SRISA, Deputy Director.
ORCID 0000-0002-3243-2751.



Galkin Valery Alekseevich ✉ – D. Sci. (Phys. & Math.), Professor, Surgut State University, Professor, Branch of SRISA, Director.
ORCID 0000-0002-9721-4026.



Parallelization of a Numerical Algorithm for Solving the Cauchy Problem for a Nonlinear Differential Equation of Fractional Variable Order Using OpenMP Technology

D. A. Tverdyi^{1,2}, R. I. Parovik^{1,2}, A. R. Hayotov³, A. K. Boltaev³*

¹ Vitus Bering Kamchatka State University,

Russia, 683032, Petropavlovsk-Kamchatsky, Pogranichnaya st., 4.

² Institute for Cosmophysical Research and Radio Wave Propagation FEB RAS,

Russia, 684034, Paratunka, Mirnaya st., 7.

³ V.I. Romanovskiy Institute of Mathematics, Uzbekistan Academy of Sciences,

100174, Tashkent, University st., 9.

Abstract. The article presents a software implementation of a parallel efficient and fast computational algorithm for solving the Cauchy problem for a nonlinear differential equation of a fractional variable order. The computational algorithm is based on a non-local explicit finite-difference scheme, taking into account the approximation of the Gerasimov-Caputo fractional derivative VO included in the main differential equation. The algorithms for parallelization of the non-local explicit finite difference scheme were implemented as functions of the user library of the C programming language using the OpenMP technology. The OpenMP technology allows implementing parallel algorithms for working with the CPU computing node using its multithreading. The C language was chosen because of its versatility and lack of strict restrictions on memory handling. Further in the paper, the efficiency of the parallel algorithm is investigated. Efficiency is understood as the optimal ratio in coordinates: acceleration of calculations – the amount of RAM memory occupied, in comparison with the sequential version of the algorithm. The average computation time is analyzed in terms of: running time, acceleration, efficiency and cost of the algorithm. These algorithms were run on two different computing systems: a gaming laptop and a computing server. For a non-local explicit scheme, a significant performance increase of 3-5 times is shown for various methods of software implementation.

Key words: fractional derivatives, heredity, memory effect, finite difference schemes, parallel computing, OpenMP


Received: 02.06.2023; Revised: 09.06.2023; Accepted: 20.06.2023; First online: 30.06.2023

For citation. Tverdyi D. A., Parovik R. I., Hayotov A. R., Boltaev A. K. Parallelization of a numerical algorithm for solving the Cauchy problem for a nonlinear differential equation of fractional variable order using OpenMP technology. *Vestnik KRAUNC. Fiz.-mat. nauki.* 2023, **43**: 2, 87-110. EDN: WFDGQO. <https://doi.org/10.26117/2079-6641-2023-43-2-87-110>.

Funding. This research was funded by grant of the President of the Russian Federation grant number MD-758.2022.1.1 on the topic “Development of mathematical models of fractional dynamics in order to study oscillatory processes and processes with saturation”.

Competing interests. There are no conflicts of interest regarding authorship and publication.

Contribution and Responsibility. All authors contributed to this article. Authors are solely responsible for providing the final version of the article in print. The final version of the manuscript was approved by all authors.

*Correspondence:  E-mail: romanparovik@gmail.com

The content is published under the terms of the Creative Commons Attribution 4.0 International License

© Tverdyi D. A., Parovik R. I., Hayotov A. R., Boltaev A. K., 2023

© Institute of Cosmophysical Research and Radio Wave Propagation, 2023 (original layout, design, compilation)





Распараллеливание численного алгоритма решения задачи Коши для нелинейного дифференциального уравнения дробного переменного порядка с помощью технологии OpenMP

Д. А. Твёрдый^{1,2}, Р. И. Паровик^{1,2*}, А. Р. Хаётов³, А. К. Болтаев³

- ¹ ФГБОУ ВО Камчатский государственный университет им. Витуса Беринга, Россия, 683032, г. Петропавловск-Камчатский, ул. Пограничная, 4.
- ² ФГБУН Институт космофизических исследований и распространения радиоволн ДВО РАН, Россия, 684034, с. Паратунка, ул. Мирная, д. 7.
- ³ Институт математики имени В. И. Романовского Академии наук Республики Узбекистан, Узбекистан, 100174, г. Ташкент, ул. Университетская, д. 9.

Аннотация. В статье представлена программная реализация параллельного эффективного и быстрого вычислительного алгоритма решения задачи Коши для нелинейного дифференциального уравнения дробного переменного порядка. Вычислительный алгоритм основан на нелокальной явной конечно-разностной схеме с учетом аппроксимации дробной производной ВО Герасимова-Капуто, входящей в основное дифференциальное уравнение. Алгоритмы распараллеливания нелокальной явной конечно-разностной схемы были реализованы в виде функций пользовательской библиотеки языка программирования C с использованием технологии OpenMP. Технология OpenMP позволяет реализовывать параллельные алгоритмы для работы с вычислительным узлом CPU, используя его многопоточность. Язык C выбран из-за его универсальности и отсутствия в нем строгих ограничений при работе с памятью. Далее в работе исследуется эффективность параллельного алгоритма. Под эффективностью понимается оптимальное соотношение в координатах: ускорение вычислений – объём занимаемой RAM памяти, по сравнению с последовательной версией алгоритма. Анализируется среднее время вычисления в терминах: время работы, ускорение, эффективность и стоимость алгоритма. Данные алгоритмы были запущены на двух различных вычислительных системах: игровом ноутбуке и вычислительном сервере. Для нелокальной явной схемы показан существенный прирост производительности в 3-5 раз при различных методах программной реализации.

Ключевые слова: дробные производные, эрмитарность, эффект памяти, явные конечно-разностные схемы, параллельные вычисления, OpenMP

Получение: 02.06.2023; Исправление: 09.06.2023; Принятие: 20.06.2023; Публикация онлайн: 30.06.2023

Для цитирования. Tverdyi D. A., Parovik R. I., Hayotov A. R., Boltaev A. K. Parallelization of a numerical algorithm for solving the Cauchy problem for a nonlinear differential equation of fractional variable order using OpenMP technology // Вестник КРАУНЦ. Физ.-мат. науки. 2023. Т. 43. № 2. С. 87-110. EDN: WFDGQO. <https://doi.org/10.26117/2079-6641-2023-43-2-87-110>.

Финансирование. Исследования выполнены в рамках гранта Президента РФ МД-758.2022.1.1 по теме «Развитие математических моделей дробной динамики с целью исследования колебательных процессов и процессов с насыщением».

Конкурирующие интересы. Конфликтов интересов в отношении авторства и публикации нет.

Авторский вклад и ответственность. Авторы участвовали в написании статьи и полностью несут ответственность за предоставление окончательной версии статьи в печать.

*Корреспонденция: ✉ E-mail: romanparovik@gmail.com

Контент публикуется на условиях Creative Commons Attribution 4.0 International License



© Tverdyi D. A., Parovik R. I., Hayotov A. R., Boltaev A. K., 2023

© ИКИР ДВО РАН, 2023 (оригинал-макет, дизайн, составление)