

УДК 004.8; 519.6

<https://doi.org/10.21869/2223-1560-2025-29-3-210-221>

Подход к самообучению автоассоциативной модели нейронной сети высокого порядка

А. В. Малышев¹ ✉

¹ Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

✉ e-mail: alta76@yandex.ru

Резюме

Цель исследования. Разработка максимально эффективного, в контексте современных технических средств реализации интеллектуальных систем, предназначенных, в т.ч., для решения задач распознавания образов, алгоритма самообучения двухслойной модели нейронной сети, который будет основан на наращивании числа нейронов и вариации весовых коэффициентов синаптических связей, с возможностью распространения его на многосвязную нейронную сеть высокого порядка с внутренним произведением векторов.

Методы. Для решения поставленной задачи в данной работе предложен подход к синтезу многосвязной модели нейронной сети высокого порядка с внутренним произведением векторов, а также алгоритм самообучения такой нейронной сети, предусматривающий оперативную коррекцию элементов матрицы эталонов вместо традиционной вариации весовых коэффициентов синаптических связей с целью снижения ресурсоёмкости выполняемых операций.

Результаты. Предложенный метод был реализован в виде программного приложения в привязке к самообучению нейронной сети высокого порядка на звукотипах речевых сигналов, представленных в растровом виде с предварительной их сегментацией из общего потока и преобразованием в полярных координатах с целью удобства обработки и хранения полученных образов в качестве обучающей выборки.

Заключение. Разработанный алгоритм в ходе проведения тестирования его программной реализации, за счёт исключения ресурсоёмкой операции вариации весовых коэффициентов и заменой её на непосредственную коррекцию матрицы эталонов, показал достаточно высокую эффективность, сходимость за конечное число шагов, обусловленную ограничением количества кодов первого приближения эталонных векторов, а также заметное быстроедействие по сравнению с известными аналогами.

Ключевые слова: нейронная сеть; звукотип; матрица эталонов; весовые коэффициенты; вектор активации.

Конфликт интересов: Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Малышев А. В. Подход к самообучению автоассоциативной модели нейронной сети высокого порядка // Известия Юго-Западного государственного университета. 2025; 29(3): 210-221. <https://doi.org/10.21869/2223-1560-2025-29-3-210-221>.

Поступила в редакцию 21.07.2025

Подписана в печать 28.08.2025

Опубликована 30.09.2025

An approach to self-training of a high-order autoassociative neural network model

Aleksandr V. Malyshev¹ ✉

¹ Southwest State University

50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

✉ e-mail: alta76@yandex.ru

Abstract

Purpose of research. To develop a self-learning algorithm for a two-layer neural network model that is as efficient as possible, given the current technical implementation of intelligent systems, including those designed for solving pattern recognition problems. This algorithm will be based on increasing the number of neurons and varying the weight coefficients of synaptic connections, with the possibility of extending it to a high-order multiconnected neural network with an internal product of vectors.

Methods. To solve this problem, this paper proposes an approach to synthesizing a high-order multiconnected neural network model with an internal product of vectors, as well as a self-learning algorithm for such a neural network. This algorithm provides for the rapid correction of the elements of the reference matrix, instead of the traditional variation of the weight coefficients of synaptic connections, in order to reduce the resource intensity of the performed operations.

Results. The proposed method was implemented as a software application linked to the self-training of a high-order neural network using speech sound types represented in raster format, pre-segmented from the general stream and transformed into polar coordinates for ease of processing and storing the resulting images as a training set.

Conclusion. The developed algorithm, during software implementation testing, demonstrated relatively high efficiency by eliminating the resource-intensive operation of varying weight coefficients and replacing it with direct correction of the reference matrix. This algorithm demonstrated relatively high efficiency, convergence in a finite number of steps due to the limited number of first-approximation codes of reference vectors, and noticeable performance compared to known analogs.

Keywords: neural network; sound type; reference matrix; weighting coefficients; activation vector.

Conflict of interest: The Author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation. Malyshev A. V. An approach to self-training of a high-order autoassociative neural network model. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2025; 29(3): 210-221 (In Russ.). <https://doi.org/10.21869/2223-1560-2025-29-3-210-221>.

Received 21.07.2025

Accepted 28.08.2025

Published 30.09.2025

Введение

Известные алгоритмы самообучения моделей нейронных сетей с внешним произведением векторов основаны на ва-

риации весовых коэффициентов синаптических связей [1, 2]. В современных моделях нейронных сетей высокого порядка с внутренним произведением векторов, обладающих высокой информа-

ционной ёмкостью при хранении сильнокоррелированных образов, весовые коэффициенты не хранятся в памяти, а вычисляются косвенно при имитации переключения нейронной сети в результате двух последовательных умножений векторов на одну и ту же матрицу эталонных векторов [3, 4]. Математическое описание модели такой нейронной сети высокого порядка в режиме имитации переключения после снятия входного вектора активизации имеет вид:

$$\begin{aligned} C &= (X^* \alpha) \parallel V \parallel, \\ Y &= C^{(n)} \parallel V \parallel^t, \\ x_i &= \text{sign}(y_i + \theta_i), \\ i &= \overline{1, N}, x_i \in \{+1, -1\}, \end{aligned} \quad (1)$$

где X – N -мерная строка компонент входного вектора; Y – N -мерная строка компонент вектора мембранного потенциала; x_i – i -я компонента входного вектора; y_i – i -я компонента вектора мембранного потенциала; α – N -мерная строка компонент вектора коэффициентов взвешивания компонент входного вектора; $X^* \alpha$ – покомпонентное произведение векторов; $\parallel V \parallel$ – прямоугольная матрица размером $N \times M$ элементов эталонных векторов, где M – число эталонных векторов; $\parallel V \parallel^t$ – обозначение транспонированной матрицы; C – M -мерная строка компонент вектора степеней сходства входного взвешенного вектора $(X^* \alpha)$ с каждым V_j , $j = \overline{1, M}$; $C^{(n)}$ – покомпонентное возведение вектора C в степень n , где n – порядок модели нейронной сети; θ_i – порог i -го нейрона; N – число нейронов в модели нейронной сети.

Компонента α_i вектора взвешивания α равна:

$$\alpha_i = 1 + \frac{1}{M} - \left| \sum_{j=1}^M v_{ij} \right| / M, \quad (2)$$

$$i = \overline{1, N}, \quad j = \overline{1, M}, \quad v_{ji} \in \{+1, -1\},$$

где v_{ij} – i -я компонента j -го эталонного вектора V_j .

Так как в памяти данной модели нейронной сети высокого порядка хранятся только компоненты v_{ij} эталонных векторов V_j , $j = \overline{1, M}$, традиционный способ её обучения и самообучения путём вариации весовых коэффициентов, вычисляемых при известных эталонах по формуле:

$$\gamma_{ik} = \sum_{j=1}^M v_{ij} \alpha_k v_{jk}, \quad i, k = \overline{1, N} \quad (3)$$

приводит к большим потерям времени на прямое и обратное преобразование матрицы эталонов $\parallel V \parallel$ в матрицу $\parallel \gamma \parallel$ весовых коэффициентов синаптических связей с промежуточным изменением γ_{ik} по алгоритму обучения [5, 6]. Кроме того, известное выражение (3) справедливо только для нейронной сети первого порядка и существенно усложняется в нейронной сети высокого порядка при $n > 1$ [7, 8].

Материалы и методы

В связи с вышеизложенным в данной работе предлагается развитие алгоритма самообучения модели нейронной сети, основанного на наращивании числа нейронов и последующей вариации весовых коэффициентов $\{\gamma_{ik}\}$, и распро-

странение его на многосвязную модель нейронной сети высокого порядка с внутренним произведением векторов [9, 10]. Предлагаемый алгоритм отличается от известных исключением ресурсоемкой операции вариации весовых коэффициентов и заменой её на непосредственную коррекцию матрицы эталонов $\|V\|$.

Используя известный критерий подобия [11, 12] как скалярное произведение нормированных векторов активизации нейронной сети и её отклика, предлагаемый алгоритм самообучения нейронной сети высокого порядка выполняется за следующие два этапа. На первом этапе в память эталонов данной нейронной сети вносятся в качестве кодов первого приближения эталонных векторов входные векторы обучающей выборки до тех пор, пока величина скалярного произведения нормированных векторов (a_{i*}) очередного вектора обучающей выборки и вектора отклика наиболее активного i^* -го нейрона не превысит установленного порога b критерия их подобия при $0,5 < b < 1$, а именно $a_{i*} > b$, т.е. до появления успешного отклика хотя бы одного i^* -го нейрона [13, 14].

На втором этапе во всех случаях, когда $a_{i*} > b$, выполняется коррекция компонент v_{ji} , накопленных в памяти кодов первого приближения эталонных векторов без дальнейшего увеличения их количества m . При этом учитывается число p_{i*} успешных откликов i^* -го наиболее активного нейрона за всё время самообучения на основе проверки двух систем неравенств:

$$a_{\bar{a}j} > \frac{p_{i*}}{(1 + p_{i*})}, \quad j = \overline{1, M}$$

$$a_{ci} > \frac{p_{i*}}{(1 + p_{i*})}, \quad j = \overline{1, N},$$

где $a_{\bar{a}j}$ – критерий подобия скалярного произведения j -го кода первого приближения и данного вектора обучающей выборки;

a_{ci} – критерий подобия скалярного произведения нормированных векторов i^* -го столбца и другого i -го столбца матрицы накопленных кодов первого приближения в пределах подмножества строк, удовлетворяющих первому неравенству.

Тем самым выделяются ji -е элементы матрицы $\|V\|$ эталонных векторов, подлежащие коррекции на данном шаге самообучения, а их значения заменяются значениями i -х компонент x_i данного вектора обучающей выборки.

Каждый шаг самообучения на каждом очередном векторе обучающей выборки начинается с образования вектора отклика X^o всей замкнутой модели нейронной сети (1) на данных X входного вектора. В качестве претендентов на наибольшую активность выбираются те нейроны, на выходе которых были образованы положительные компоненты вектора отклика X^o . Наиболее активный i^* -й нейрон выделяется из них путём:

– последовательного опроса разомкнутой модели нейронной сети однокомпонентными входными векторами вида:

$$X_{li} = (0 \dots 0 \ x_i^0 \ 0 \dots 0) \text{ при } x_i^0 > 0; \quad (4)$$

– нахождения по каждому из них выходного вектора отклика Y_i разомкнутой модели;

– выделения в их последовательности $\{Y_i\}$ одного i^* -го, наиболее близкого по скалярному произведению к данному вектору обучающей выборки:

$$a_{i^*} = \max_{\text{по } i} \{a_i\} = \max_{\text{по } i} \left\{ \frac{X}{X^\mu} \frac{Y_i}{Y_i^\mu} \right\}. \quad (5)$$

где XY_i – скалярное произведение векторов;

X^μ – длина вектора, вычисляемая по формуле

$$X^\mu = \sqrt{\sum_{i=1}^N x_i^2}. \quad (6)$$

Тем самым стимулируется повышение адекватности ответов наиболее активного нейрона на близкие векторы обучающей выборки, принадлежащие одному и тому же классу образов, а в ходе самообучения зарождается и подкрепляется центр активности данного класса образов, присутствующих во входной последовательности вышеозначенных векторов [15, 16]. Причём в памяти, хранящей матрицу $\|V\|$ корректируемых эталонных векторов, может быть создано большое количество центров активности сильно-коррелированных образов, которое, благодаря повышению порядка модели нейронной сети и взвешиванию компонент входного вектора, существенно превышает возможности модели Хопфилда.

Словесная формулировка одного шага алгоритма самообучения модели нейронной сети высокого порядка на

одном очередном входном векторе обучающей выборки:

1. Определяется и запоминается вектор отклика X^o замкнутой модели нейронной сети на данный входной вектор X обучающей выборки.

2. Выделяется подмножество нейронов- претендентов на наибольшую активность по критерию: $x_i^o > 0$.

3. Находится наиболее активный i^* -й нейрон путём последовательного опроса разомкнутой модели нейронной сети однокомпонентными входными векторами вида (4), определения по каждому из них выходного вектора Y_i и выделения в их последовательности $\{Y_i\}$ одного i^* -го, наиболее близкого к данному вектору обучающей выборки по максимуму множества скалярных произведений нормированных векторов (5, 6).

4. Проверяется установленный порог критерия подобия: $a_{i^*} \leq b$. Если $a_{i^*} > b$, переход к п. 6.

5. В память модели записывается данный вектор обучающей выборки X в качестве очередного эталонного вектора матрицы эталонов $\|V\|$, а затем переход к п. 12.

6. Увеличивается на единицу число p_{i^*} успешных откликов i^* -го наиболее активного нейрона:

$$p_{i^*} := p_{i^*} + 1.$$

7. Вычисляется вектор скалярного произведения данного вектора обучающей выборки и всех m накопленных кодов первого приближения в матрице $\|V\|$ эталонов:

$$A_j^t = (a_{j1}, \dots, a_{jm})^t = \left\| \frac{V_j}{V_j^\mu} \left\| \left[\frac{X\alpha}{(X\alpha)^\mu} \right]^t \right\| \right\|,$$

где первый множитель – это матрица $\|V\|$ накопленных к данному шагу обучения кодов первого приближения, представленная в виде столбца нормированных эталонных векторов-строк; m – число кодов первого приближения, накопленных в матрице $\|V\|$ к данному шагу обучения.

8. Выделяется подмножество строк $\{j1\}$ матрицы $\|V\|$, элементы которых целесообразно варьировать в данном шаге обучения по критерию:

$$a_{j1} > \frac{p_{i^*}}{(1 + p_{i^*})}, j1 \in \{j\}. \quad (7)$$

9. Вычисляются критерии подобия i^* -го столбца выделенного в п. 8 подмножества строк $\{j1\}$ матрицы $\|V\|$ с другими столбцами этого же подмножества строк как их скалярное произведение:

$$a_{ci} = \frac{1}{\sqrt{\sum_{j1} (v_{j1i^*})^2}} \frac{1}{\sqrt{\sum_{j1} (v_{j1i})^2}} \times \\ \times (v_{1i}, \dots, v_{j1i}, \dots, v_{mi}) (v_{1i^*}, \dots, v_{j1i^*}, \dots, v_{mi^*}), \\ j1 \in \{j\}, \quad i \neq i^*, \quad i = \overline{1, N}.$$

10. Выделяется подмножество столбцов $\{i1\}$ матрицы $\|V\|$, элементы которых целесообразно варьировать в пределах подмножества $\{j1\}$ в данном шаге обучения по критерию:

$$a_{ci1} > \frac{p_{i^*}}{(1 + p_{i^*})}, i1 \in \{i\}. \quad (8)$$

11. Корректируется $j1i1$ -е элементы матрицы $\|V\|$ путем замены их на $i1$ -е

компоненты данного вектора обучающей выборки:

$$v_{j1i1} := x_{i1}.$$

12. Вычисляется новый вектор α коэффициентов взвешивания компонент входного вектора по элементам модифицированной матрицы эталонов $\|V\|$ и замещается старый вектор α предыдущего шага обучения по формуле (2), где

– $M = m$ при переходе по п.п. 11, 12;

– $M = m + 1$ при переходе по п.п. 5, 12.

13. Конец шага обучения.

Результаты и их обсуждение

Фрагмент кода [17, 18] представленного алгоритма, реализованного на языке программирования Java, приведен на рис.1. Сходимость представленного алгоритма обусловлена ограничением числа m кодов первого приближения эталонных векторов, накапливаемых в памяти модели нейронной сети на первом этапе, а также постепенным уменьшением степени модификации элементов матрицы $\|V\|$ без изменения числа m . На втором этапе по мере увеличения числа p_{i^*} возрастают длины обучающих выборок, а выражение в правой части соответствующих неравенств (7, 8) при большом числе шагов обучения выглядит следующим образом:

$$p_{i^*}/(1 + p_{i^*}) \rightarrow 1.$$

При этом коррекция матрицы $\|V\|$ прекращается, так как значения критериев подобия в левой части неравенств не превышают единицы:

$$a_{ej} \leq 1, a_{cj} \leq 1.$$

```

// Параметры алгоритма
private static final double B = 0.8; // Порог критерия подобию
private static final double[][] V = new double[/* количество эталонов */][/* размер
входного вектора */]; // матрица эталонов
private static final int[] p = new int[/**/]; // массив счетчиков успешных откликов
private static final double[] alpha = new double[/**/]; // коэффициенты взвешивания
private static final double[] x = new double[/**/]; // текущий входной вектор
// Метод для нормализации вектора
private static double[] normalize(double[] vec) {
    double norm = 0.0;
    for (double v : vec) {
        norm += v * v;
    }
    norm = Math.sqrt(norm);
    double[] normalized = new double[vec.length];
    for (int i = 0; i < vec.length; i++) {
        normalized[i] = vec[i] / norm;
    }
    return normalized;
}
// Метод для вычисления скалярного произведения двух векторов
private static double dotProduct(double[] v1, double[] v2) {
    double sum = 0.0;
    for (int i = 0; i < v1.length; i++) {
        sum += v1[i] * v2[i];
    }
    return sum;
}
// Основной метод шага обучения
public static void trainStep(double[] xInput) {
// 1. Определяем и запоминаем вектор отклика X0 модели на вход X
    double[] normalizedX = normalize(xInput);
    double[] X0 = new double[V.length]; // вектор отклика
    for (int i = 0; i < V.length; i++) {
        double[] normalizedV = normalize(V[i]);
        X0[i] = dotProduct(normalizedV, normalizedX);
    }
// 2. Выделяем подмножество нейронов-претендентов с X0 > 0
    int iStar = -1;
    double maxActivation = -Double.MAX_VALUE;
    for (int i = 0; i < X0.length; i++) {
        if (X0[i] > 0 && X0[i] > maxActivation) {
            maxActivation = X0[i];
            iStar = i;
        }
    }
    if (iStar == -1) {
        return;
    }
// 3. Находим наиболее активный нейрон i* по скалярному произведению
    int selectedNeuron = -1;
    double maxScalarProduct = -Double.MAX_VALUE;
    for (int i = 0; i < V.length; i++) {
        double[] normalizedV = normalize(V[i]);
        double scalarProd = dotProduct(normalizedV, normalizedX);
        if (scalarProd > maxScalarProduct) {
            maxScalarProduct = scalarProd;
            selectedNeuron = i;
        }
    }
// 4. Проверяем критерий подобию ai* ≤ b
    double aiStar = maxScalarProduct; // предполагаем, что это мера подобию
    if (aiStar > B) {
        return;
    }
// 5. Запись входного вектора в матрицу эталонов ||V||
    int m = /* текущее число кодов */;
    p[iStar]++;
// 6. Увеличиваем на 1 счетчик p[i*]
    p[iStar]++;
// 7. Вычисляем вектор A_э^t ...

```

Рис. 1. Фрагмент кода алгоритма самообучения

Fig. 1. Fragment of the self-learning algorithm programming code

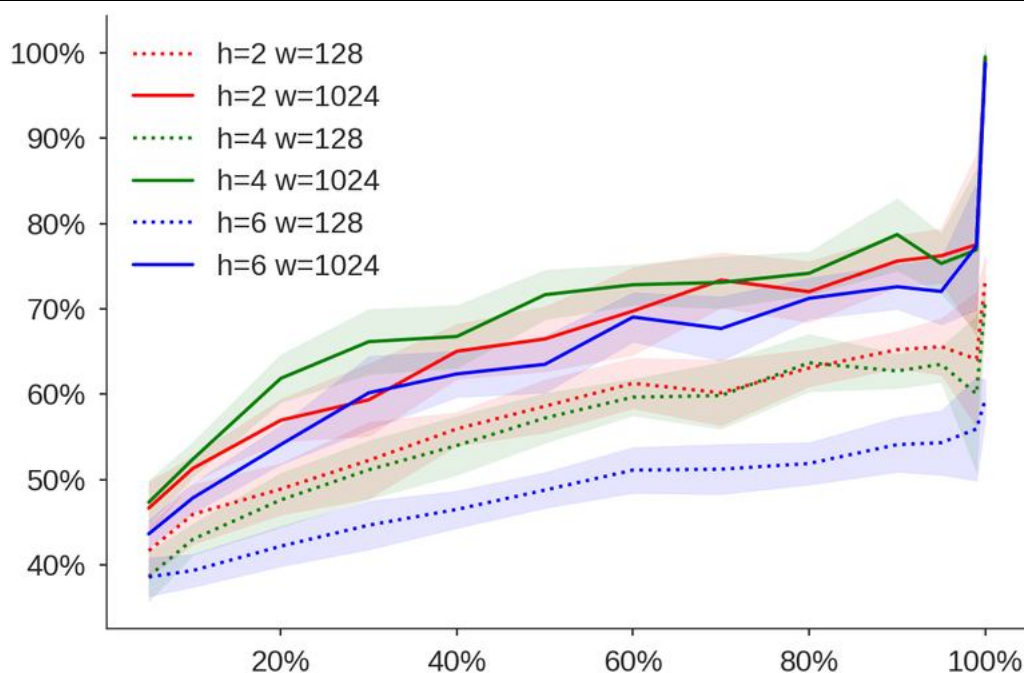


Рис. 2. Зависимость качества обучения от слоёв и объёма выборки

Fig. 2. Dependence of training quality on layers and sample size

Выводы

Сходимость проверена путём моделирования алгоритма [19, 20] на типовом ПЭВМ с процессором с максимально низкими на текущий момент характеристиками (Intel Core2 Duo E8400) при имитации обучения на звукотипах речевых сигналов, представленных в полярных координатах и в растровом формате. При относительно высоких искажениях и вариациях входного образа процесс обучения завершился за 15 шагов

на 50 входных векторах с накоплением в памяти $m=6$, принадлежащих одному классу образов (рис. 2).

Требуемая ёмкость памяти для хранения матрицы $\|V\|$ может быть уменьшена в несколько раз без снижения достоверности классификации путём повторного самообучения с использованием содержимого памяти m накопленных и откорректированных кодов первого приближения эталонных векторов в качестве обучающей выборки.

Список литературы

1. Бобырь М.В., Кулабухов С.А., Милостная Н.А. Обучение нейро-нечеткой системы на основе метода разности площадей // Искусственный интеллект и принятие решений. 2016. №4. С. 15-26.
2. Schmidhuber J. Deep learning in neural networks: an overview // Neural networks. 2015. Vol. 61. P. 85-117.

3. Емельянов С.Г., Бобырь М.В., Анциферов А.В. Алгоритм самообучения адаптивных нейро-нечетких систем на основе мягких вычислений // Информационно-измерительные и управляющие системы. 2013. №8. С. 3-9.
4. Малышев А.В., Типикин А.П., Тараненко К.Ю. Самообучение автоассоциативной модели нейронной сети высокого порядка // Известия Курского государственного технического университета. 1998. №2. С. 63-68.
5. Ashwin P., Coombes S., Nicks R. Mathematical frameworks for oscillatory network dynamics in neuroscience // Journal of mathematical neuroscience. 2016. Vol. 6. P. 1-92.
6. Malyshev A., Novikova A. Neural algorithm of reconfiguration for matrix multiprocessor // International Conference «Modern trends in education and science». Tambov, 2014. P. 10-11.
7. Бобырь М.В. Обучение нейро-нечётких систем. М.: Аргамак-Медиа, 2017. 240 с.
8. Coexisting multi-stable patterns in memristor synapse-coupled Hopfield neural network with two neurons / C. Chen, J. Chen, M. Chen, B. Bao, H. Bao // Nonlinear dynamics. 2019. Vol. 95. P. 3385-3399.
9. Малышев А.В., Алексеев А.Е., Геворкян А.Ю. Распознавание и отслеживание объектов посредством инструментов нейросетевых моделей // Программная инженерия: современные тенденции развития и применения: материалы VIII Всероссийской научно-практической конференции. Курск, 2024. С. 97-101.
10. Design and analysis of multiscroll memristive Hopfield neural network with adjustable memductance and application to image encryption / Q. Lai, ZH. Wan, H. Zhang, G. Chen // IEEE Transactions on neural networks and learning systems. 2023. Vol. 34. P. 7824-7837.
11. Danca M., Kuznetsov N. Hidden chaotic sets in a Hopfield neural system // Chaos, solitons & fractals. 2017. Vol. 103. P. 144-150.
12. Quantum Hopfield neural network / P. Rebentrost, T. Bromley, C. Weedbrook, S. Lloyd // Physical review. 2018. Vol. 98. P. 042308.
13. Малышев А.В., Глухов Н.К. Применение искусственного интеллекта для анализа жизненного цикла технологического проекта // Молодёжь и XXI век: материалы 13-й Международной научной конференции. Курск, 2024. С. 348-351.
14. Ong B., Sugiura K., Zettsu K. Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting // Neural computing & applications. 2016. Vol. 27. P. 1553-1566.
15. Malyshev A., Krylov P., Volodin O. Metaheuristic analytical instruments in optimizing the shopping center parking area: a transportation model // International journal of advanced science and technology. 2019. Vol. 28. P. 198-207.
16. Разработка интеллектуальной системы для прогнозирования рисков сердечно-сосудистых заболеваний / А.В. Малышев, Е.И. Пузырев, М.В. Прохоров, Н.Г. Нефедов // Известия Юго-Западного государственного университета. Серия: Управление, вы-

числительная техника, информатика. Медицинское приборостроение. 2022. Т. 12, № 3. С. 46-61. <https://doi.org/10.21869/2223-1536-2022-12-3-46-61>

17. Дзюбин И.А., Малышев А.В. Способы оптимизации высоконагруженных приложений // Программная инженерия: современные тенденции развития и применения: материалы V Всероссийской научно-практической конференции. Курск, 2021. С. 12-20.

18. Чаплыгин А.А., Малышев А.В. Операционная система с ядром на основе виртуальной машины с уменьшенным набором команд // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2020. Т. 10, №1. С. 8-20.

19. Малышев А.В., Родин А.А., Сабуров В.Г. Проектирование системы автоматизированного перевода для мобильных устройств // Программная инженерия: современные тенденции развития и применения: материалы Всероссийской научной конференции. Курск, 2017. С. 143-147.

20. Воробьева И.С., Псарёва А.Д., Малышев А.В. Методы семантической кластеризации текстов // Программная инженерия: современные тенденции развития и применения: материалы V Всероссийской научно-практической конференции. Курск, 2021. С. 81-84.

References

1. Bobyr M., Kulabukhov S., Milostnaya N. Training a neuro-fuzzy system based on the area difference method. *Iskusstvennyi intellect i prinyatie resheniy = Artificial Intelligence and Decision Making*. 2016; (4): 15-26. (In Russ.).

2. Schmidhuber J. Deep learning in neural networks: an overview. *Neural networks*. 2015; 61: 85-117. (In Russ.).

3. Emelianov S., Bobyr M., Antsiferov A. Self-learning algorithm for adaptive neuro-fuzzy systems based on soft computing. *Informatsionno-izmeritelnye i upravlyaushie sistemy = Information, measuring and control systems*. 2013; (8): 3-9. (In Russ.).

4. Malyshev A., Tipikin A., Taranenko K. Self-training of an autoassociative model of a high-order neural network. *Izvestiya Kurskogo gosudarstvennogo tekhnicheskogo universiteta = Proceeding of Kursk State Technical University*. 1998; (2): 63-68. (In Russ.).

5. Ashwin P., Coombes S., Nicks R. Mathematical frameworks for oscillatory network dynamics in neuroscience. *Journal of mathematical neuroscience*. 2016; 6: 1-92.

6. Malyshev A., Novikova A. Neural algorithm of reconfiguration for matrix multiprocessor. In: *International Conference «Modern trends in education and science»*. Tambov; 2014. P. 10-11.

7. Bobyr M. Training of neuro-fuzzy systems. Moscow: Argamak-Media; 2017. 240 p. (In Russ.).

8. Chen C., Chen J., Chen M., Bao B., Bao H. Coexisting multi-stable patterns in memristor synapse-coupled Hopfield neural network with two neurons. *Nonlinear dynamics*. 2019; 95: 3385-3399.

9. Malyshev A., Alekseev A., Gevorkyan A. Recognition and tracking of objects using neural network model tools. In: *Programmnyaya inzheneriya: sovremennye tendentsii razvitiya i primeneniya: materialy VIII Vserossiiskoi nauchno-prakticheskoi konferentsii = Proceedings of the VIII All-Russian scientific and practical conference «Software engineering: modern trends in development and application»*. Kursk; 2024. P. 97-101. (In Russ.).

10. Lai Q., Wan ZH., Zhang H., Chen G. Design and analysis of multiscroll memristive Hopfield neural network with adjustable memductance and application to image encryption. *IEEE Transactions on neural networks and learning systems*. 2023; 34: 7824-7837.

11. Danca M., Kuznetsov N. Hidden chaotic sets in a Hopfield neural system. *Chaos, solitons & fractals*. 2017; 103: 144-150.

12. Rebentrost P., Bromley T., Weedbrook C., Lloyd S. Quantum Hopfield neural network. *Physical review*. 2018; 98: 042308.

13. Malyshev A., Glukhov N. Application of artificial intelligence for the analysis of the life cycle of a technological project. In: *Molodezh' i XXI vek: materialy 13-i Mezhdunarodnoi nauchnoi konferentsii = Proceedings of the 13th International Scientific Conference «Youth and the 21st Century»*. Kursk; 2024. P. 348-351. (In Russ.).

14. Ong B., Sugiura K., Zettsu K. Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting. *Neural computing & applications*. 2016; 27: 1553-1566.

15. Malyshev A., Krylov P., Volodin O. Metaheuristic analytical instruments in optimizing the shopping center parking area: a transportation model. *International journal of advanced science and technology*. 2019; 28: 198-207.

16. Malyshev A.V., Puzyrev E.I., Prokhorov M.V., Nefedov N.G. The Program of the Cardiovascular Disease Risk Prediction System. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie = Proceedings of the Southwest State University. Series: Control, Computing Engineering, Information Science. Medical Instruments Engineering*. 2022;12(3):46-61. (In Russ.). <https://doi.org/10.21869/2223-1536-2022-12-3-46-61>

17. Dzyubin I., Malyshev A. Methods for optimizing highly loaded applications. In: *Materialy V Vserossiiskoi nauchno-prakticheskoi konferentsii «Programmnyaya inzheneriya: sovremennye tendentsii razvitiya i primeneniya» = Proceedings of the V All-Russian scientific and practical conference «Software engineering: modern trends in development and application»*. Kursk; 2021. P. 12-20. (In Russ.).

18. Chaplygin A., Malyshev A. An operating system with a kernel based on a virtual machine with a reduced instruction set. *Izvestiya Yugo-Zapadnogo gosudarstvennogo univer-*

siteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie = *Proceedings of the Southwest State University. Series: Control, Computing Engineering, Information Science. Medical Instruments Engineering*, 2020; 10(1): 8-20. (In Russ.).

19. Malyshev A., Rodin A., Saburov V. Design of an automated translation system for mobile devices. In: *Materialy Vserossiiskoi nauchnoi konferentsii «Programmnaya inzheneriya: sovremennye tendentsii razvitiya i primeneniya» = Proceedings of the All-Russian scientific and practical conference «Software engineering: modern trends in development and application»*. Kursk; 2017. P. 143-147. (In Russ.).

20. Vorobyova I., Psareva A., Malyshev A. Methods of semantic clustering of texts. In: *Materialy V Vserossiiskoi nauchno-prakticheskoi konferentsii «Programmnaya inzheneriya: sovremennye tendentsii razvitiya i primeneniya» = Proceedings of the V All-Russian scientific and practical conference «Software engineering: modern trends in development and application»*. Kursk; 2021. P. 81-84. (In Russ.).

Информация об авторе / Information about the Author

Малышев Александр Васильевич, кандидат технических наук, доцент, завкафедрой программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: alta76@yandex.ru

Aleksandr V. Malyshev, Cand. of Sci. (Engineering), Associate Professor, Head of the Software Engineering Department, Southwest State University, Kursk, Russian Federation, e-mail: alta76@yandex.ru