

# ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО  
ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА

2017  
Т. 6, № 3

ISSN 2305-9052

СЕРИЯ

## «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий,  
в которых должны быть опубликованы результаты диссертаций  
на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение  
высшего образования «Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Математическое и программное обеспечение высокопроизводительных вычислительных систем
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

### Редакционная коллегия

**Л.Б. Соколинский**, д.ф.-м.н., проф., *гл. редактор*

**В.П. Танана**, д.ф.-м.н., проф., *зам. гл. редактора*

**М.Л. Цымблер**, к.ф.-м.н., доц., *отв. секретарь*

**Г.И. Радченко**, к.ф.-м.н., доц.

**А.С. Порозов**, *техн. секретарь*

### Редакционный совет

**С.М. Абдуллаев**, д.г.н., профессор

**А. Андреяк**, PhD, профессор (Германия)

**В.И. Бердышев**, д.ф.-м.н., акад. РАН, *председатель*

**В.В. Воеводин**, д.ф.-м.н., чл.-кор. РАН

**Дж. Донгарра**, PhD, профессор (США)

**С.В. Зыкин**, д.т.н., профессор

**Д. Маллманн**, PhD, профессор (Германия)

**А.В. Панюков**, д.ф.-м.н., профессор

**Р. Продан**, PhD, профессор (Австрия)

**А.Н. Томилин**, д.ф.-м.н., профессор

**В.Е. Третьяков**, д.ф.-м.н., чл.-кор. РАН

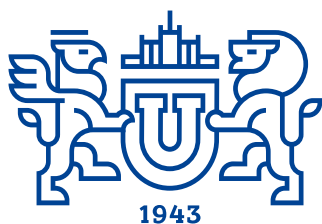
**В.И. Ухоботов**, д.ф.-м.н., профессор

**В.Н. Ушаков**, д.ф.-м.н., чл.-кор. РАН

**М.Ю. Хачай**, д.ф.-м.н., профессор

**А. Черных**, PhD, профессор (Мексика)

**П. Шумяцкий**, PhD, профессор (Бразилия)



# BULLETIN

**OF THE SOUTH URAL STATE UNIVERSITY** 2017  
vol. 6, no. 3

SERIES

“COMPUTATIONAL  
MATHEMATICS AND SOFTWARE  
ENGINEERING”

ISSN 2305-9052

---

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.  
Seriya “Vychislitel'naya Matematika i Informatika”

---

## South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- High performance computing
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

### Editorial Board

**L.B. Sokolinsky**, South Ural State University (Chelyabinsk, Russia)  
**V.P. Tanana**, South Ural State University (Chelyabinsk, Russia)  
**M.L. Zymbler**, South Ural State University (Chelyabinsk, Russia)  
**G.I. Radchenko**, South Ural State University (Chelyabinsk, Russia)  
**A.S. Porozov**, South Ural State University (Chelyabinsk, Russia)

### Editorial Council

**S.M. Abdullaev**, South Ural State University (Chelyabinsk, Russia)  
**A. Andrzejak**, Heidelberg University (Germany)  
**V.I. Berdyshev**, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)  
**J. Dongarra**, University of Tennessee (USA)  
**M.Yu. Khachay**, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)  
**D. Mallmann**, Julich Supercomputing Centre (Germany)  
**A.V. Panyukov**, South Ural State University (Chelyabinsk, Russia)  
**R. Prodan**, University of Innsbruck (Innsbruck, Austria)  
**P. Shumyatsky**, University of Brasilia (Brazil)  
**A. Tchernykh**, CICESE Research Center (Mexico)  
**A.N. Tomilin**, Institute for System Programming of the RAS (Moscow, Russia)  
**V.E. Tretyakov**, Ural Federal University (Yekaterinburg, Russia)  
**V.I. Ukhobotov**, Chelyabinsk State University (Chelyabinsk, Russia)  
**V.N. Ushakov**, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)  
**V.V. Voevodin**, Lomonosov Moscow State University (Moscow, Russia)  
**S.V. Zykin**, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Omsk, Russia)

# Содержание

## Вычислительная математика

A REGULARIZED LEVENBERG–MARQUARDT TYPE METHOD APPLIED TO THE STRUCTURAL INVERSE GRAVITY PROBLEM IN A MULTILAYER MEDIUM AND ITS PARALLEL REALIZATION A.F. Skurydina .....	5
---	---

## Дискретная математика и математическая кибернетика

PREIMAGE ATTACK ON MD4 HASH FUNCTION AS A PROBLEM OF PARALLEL SAT-BASED CRYPTANALYSIS I.A. Gribanova, O.S. Zaikin, I.V. Otpuschennikov, A.A. Semenov .....	16
---	----

## Информатика, вычислительная техника и управление

ОБЗОР МЕТОДОВ ОБУЧЕНИЯ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ А.В. Созыкин .....	28
АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ НЕЧЕТКИХ ПРАВИЛ ДЛЯ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ С ГУСЕНИЧНЫМ ШАССИ НА ОСНОВЕ ЧИСЛОВЫХ ДАННЫХ Е.А. Пташко, В.И. Ухоботов .....	60
ВЕКТОРНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ТЕРМОВ Д.В. Бондарчук .....	73
ОБНАРУЖЕНИЕ И ЛОКАЛИЗАЦИЯ СТРОЕНИЙ ЗАДАННОЙ ФОРМЫ НА АЭРОСНИМКАХ В ИНФРАКРАСНОМ ДИАПАЗОНЕ А.В. Дунаева, Ф.А. Корнилов .....	84

# Contents

## Computational Mathematics

A REGULARIZED LEVENBERG–MARQUARDT TYPE METHOD APPLIED TO THE STRUCTURAL INVERSE GRAVITY PROBLEM IN A MULTILAYER MEDIUM AND ITS PARALLEL REALIZATION A.F. Skurydina .....	5
---	---

## Discrete mathematics and mathematical cybernetics

PREIMAGE ATTACK ON MD4 HASH FUNCTION AS A PROBLEM OF PARALLEL SAT-BASED CRYPTANALYSIS I.A. Gribanova, O.S. Zaikin, I.V. Otpuschennikov, A.A. Semenov .....	16
---	----

## Informatics, computers and control

AN OVERVIEW OF METHODS FOR DEEP LEARNING IN NEURAL NETWORKS A.V. Sozykin .....	28
AUTOMATIC GENERATION OF FUZZY RULES FOR CONTROL OF A MOBILE ROBOT WITH TRACK CHASIS BASED ON NUMERICAL DATA E.A. Ptashko, V.I. Ukhobotov .....	60
VECTOR SPACE MODEL OF KNOWLEDGE REPRESENTATION BASED ON SEMANTIC RELATEDNESS D.V. Bondarchuk .....	73
SPECIFIC SHAPE BUILDING DETECTION FROM AERIAL IMAGERY IN INFRARED RANGE A.V. Dunaeva, F.A. Kornilov .....	84



This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

# A REGULARIZED LEVENBERG–MARQUARDT TYPE METHOD APPLIED TO THE STRUCTURAL INVERSE GRAVITY PROBLEM IN A MULTILAYER MEDIUM AND ITS PARALLEL REALIZATION\*

© 2017 A.F. Skurydina

*Krasovskii Institute of Mathematics and Mechanics Ural Branch of RAS  
(S.Kovalevskaya str. 16, Yekaterinburg, 620990 Russia),*

*Ural Federal University named after the first President of Russia B.N.Yeltsin  
(Mira str. 19, Yekaterinburg, 620002 Russia)*

*E-mail: afinapal@gmail.com*

Received: 05.06.2017

The structural inverse gravity problem in a multilayer medium is one of the most important geophysics problem. Until recently, the problem was reduced to the separation of gravitational fields and the restoration of unknown layers independently. Now the methods are in demand that allow find unknown layers simultaneously. For solving Urysohn integral equation of the first kind describing the problem regularized algorithms Levenberg–Marquardt type with weight factors are investigated. A new Levenberg–Marquardt type method based on Levenberg–Marquardt scheme is proposed. A regularized Levenberg–Marquardt type method compared with classic Levenberg–Marquardt method. For classic Levenberg–Marquardt method some computational optimizations are offered. The numerical experiments using model gravitational data allow to compare convergence rates, relative errors and program execution times of classic Levenberg–Marquardt algorithm and Levenberg–Marquardt method. The parallel programs implementing the algorithms are developed using CUDA and OpenMP technologies.

*Keywords: Tikhonov regularization scheme, integral Urysohn type equation of first kind, regularized Levenberg–Marquardt method, regularized Levenberg–Marquardt type method, inverse gravimetry multilayer problem.*

## FOR CITATION

Skurydina A.F. A Regularized Levenberg–Marquardt Type Method Applied to the Structural Inverse Gravity Problem in a Multilayer Medium and its Parallel Realization. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 5–15. DOI: 10.14529/cmse170301.

## Introduction

This paper is concerned with iterative solutions solving the inverse structural gravity problem in a multilayer medium and is a continuation of the series of works [1, 2].

Hence we consider an operator equation

$$A(u) = f, \tag{1}$$

where  $A(u)$  is nonlinear Frechet differentiable integral Urysohn type operator between Hilbert spaces  $U, F$ ,  $u = (u_0, \dots, u_L)$  are unknown functions describing  $L$  desired interfaces,  $f$  is the total gravitational field. The solution of (1) does not depend continuously on the data and thus using of noise-contaminated data would lead to a meaningful deviation from solution. Hence a stable

\* The paper is recommended for publication by the Program Committee of the International Scientific Conference “Parallel Computational Technologies (PCT) 2017”.

solution of (1) requires regularization techniques, for example the method of Tikhonov. We obtain

$$A'(u)^*(A(u) - f_\delta) + \alpha(u - u^0) = 0, \quad (2)$$

where  $A'(u)^*$  is a conjugated operator for derivative operator  $A'(u)$ ,  $\alpha > 0$  is a regularization parameter,  $\|f - f_\delta\| \leq \delta$ ,  $u^0$  is an initial approximation. So we will assume solving equation (2). To solve (2) the regularized Levenberg–Marquardt algorithm can be used [3]:

$$u^{k+1} = u^k - \gamma[A'(u^k)^*A'(u^k) + \alpha I]^{-1}[A'(u^k)^*(A(u^k) - f_\delta) + \alpha(u - u^0)] \quad (3)$$

where  $\gamma$  is the damping factor. This method used in iterative solution nonlinear inverse problems of filtration, borehole and exploration geophysics ([3–5]) etc. In the article [6] the method (3) strong convergence to the solution is set up for the Tikhonov-regularized equation on the assumption that the condition of the sourcewise representability of the solution  $z$  of the equation (1) and the Lipschitz conditions for the derivative of the operator  $A$  are fulfilled and the initial approximation is taken from a rather small neighborhood of the regularized solution.

$$\|A'(u)\| \leq N_1, \quad \|A'(u) - A'(v)\| \leq N_2\|u - v\|,$$

$$z - \xi = A'(z)^*v, \quad \|v\| < 1/N_2.$$

This method is complex to implement. It takes a lot of time for matrix to matrix multiplication, matrix inversion. It is possible to use iterative methods for matrix inversion, so the iterative process is two-step: at each step we reduce the problem to SLAE, which we solve by some iterative method. We can see that LM algorithm tends to have larger computational overheads with an increase in the size of input data.

The previous work [2] is concerned with a regularized Levenberg–Marquardt method (CLM) which within the weight factors approach proposed in [7] lets find simultaneously several structural boundaries described by unknown functions  $u_0, \dots, u_L$  in equation (1) using the total gravitational field  $f$ . Weight factors  $w_i$  will be chosen as follows:

$$\begin{aligned} F &= [F_1, F_2, \dots, F_L] = (f_1, f_2, \dots, f_{M \times L}, \dots, f_{L \times M \times N}) \\ &\rightarrow (w_1, w_2, \dots, w_{L \times M \times N}), \\ w_i &= \frac{|f_i|^\beta}{\max_i |f_i|^\beta}, \quad \beta > 1, \end{aligned} \quad (4)$$

where  $F_l (l = 1, 2, \dots, L)$  are anomalous fields generated by the gravitating mass located below the corresponding depths  $H_l$  for the sought surfaces of interface  $S_l (l = 1, 2, \dots, L)$ . Weight factors depend on field  $F_l$  which separated from field original  $F$  using preliminary processing of gravity observations [8].

Linearized gradient type methods based on linearized steepest descent method with weight factors (5) for solving the gravity problem are considered in works [9, 10]

$$u_i^{k+1} = u_i^k - \psi w_i \frac{\|S(u^k)\|^2}{\|A'(u^k)S(u^k)\|^2} S_i(u^k), \quad (5)$$

where  $S(u^k) = A'(u^k)^*(A(u^k) - f)$ ,  $\psi$  is damping factor. This method is suited to deal with multilayer problem but there is a matrix to matrix multiplication operation.

Also in [11] a Levenberg–Marquardt gradient method based on Landweber-type scheme is proposed

$$u_i^{k+1} = u_i^k - \psi \left[ \frac{(A_i[u^k] - F_i)}{\|\nabla A_i[u^k]\|^2} \frac{\partial A_i[u^k]}{\partial u_i^k} \right].$$

As seen, this method is fast but is suitable only for finding interfaces in two layer model.

The present paper is focused on comparison of relative errors, numbers of iterations and computation times between classic regularized Levenberg–Marquardt method (LM) and CLM. Here there are used gravitational field models with uniform 15% noise. In a view of big memory consumption and high computational complexity of LM some algorithmic optimizations are proposed. On a basis of algorithms the parallel programs are implemented using OpenMP and CUDA technologies. The performance estimations of parallel programs are obtained.

The rest of the paper is organized as follows. The section 1 is dedicated to inverse multilayer gravity problem definition. The section 2 devoted to LM and CLM description. The next section 3 describes a techniques and principles used for program development. The section 4 presents the numerical results using quasi-model gravitational data and the results of parallel implementations. The final section lists the conclusions.

## 1. Multilayer structural gravity problem statement

The three-dimensional structural inverse gravity problem on finding interfaces between medium layers on the basis of data on the gravitational field measured in a certain area of the earth surface, and the density jumps.

It is assumed that the lower half-space consists of several layers with a constant density  $\Delta\sigma_l (l = 1, \dots, L)$ , divided by desired interfaces  $S_l$ , where  $L$  is the number of interfaces (fig. 1). The gravitational effect of such a half-space is equal to the sum of the gravitational effects of all the interfaces.

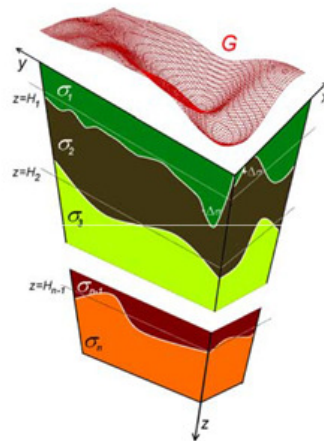


Fig. 1. Model of multilayer medium

Let the interfaces be described by the equations  $u_l = u_l(x, y)$  and the jumps of density are equal to  $\Delta\sigma_l$ . The interfaces have horizontal asymptotic planes  $u_l = H_l$ , i.e.

$$\lim_{|x|, |y| \rightarrow \infty} |u_l(x, y) - H_l| = 0.$$

Functions  $u_l = u_l(x, y)$  describing the desired interfaces satisfy operator equation (2), operator  $A$  takes the form

$$A(u) = \sum_{l=1}^L f \Delta \sigma_l \frac{1}{4\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + u_l^2(x, y)]^{1/2}} - \frac{1}{[(x-x')^2 + (y-y')^2 + H_l^2]^{1/2}} \right\} = \Delta g(x', y'), \quad (6)$$

where  $f$  is the gravitational constant,  $\Delta \sigma_l (l = 1, \dots, L)$  is the density jump,  $\Delta g(x', y') = \sum_{l=1}^L g_l$  is the sum of an anomalous gravitational fields. Preliminary processing of the gravity data with the aim to select the anomalous field from the measured gravity data is performed using the methodology [8]. The problem is undetermined because of attemption to find several unknown functions  $u_l = u_l(x, y)$  from the given function  $\Delta g(x', y')$ . So it's necessary to use the weight factors which can be found from formula [7].

## 2. Numerical methods for solving the problem

To solve (6) the regularized Levenberg–Marquardt algorithm with weight factors can be used:

$$u^{k+1} = u^k - \gamma [A'(u^k)^* A'(u^k) + \alpha I]^{-1} \Lambda [A'(u^k)^* (A(u^k) - f_\delta) + \alpha (u - u^0)], \quad (7)$$

where  $\Lambda$  is operator with a corresponding diagonal matrix with the weight factors on the main diagonal.

*Remark.* In nonlinear inverse gravimetry problems in a discrete representation the matrix  $A'(u^k)$  is ill-conditioned which entails significant increasing the condition number of  $A'(u^k)^* A'(u^k)$ .

The second method is a Levenberg–Marquardt regularized Levenberg–Marquardt algorithm [2]. Here iterative process approximates each of the solution components  $u_l, l = 1, \dots, L$ :

$$u_l^{k+1} = u_l^k - \gamma \frac{1}{\varphi_l} \Lambda [A'(u_l^k)^* (A(u^k) - f_\delta) + \alpha (u_l^k - u_l^0)], \quad (8)$$

where

$$\varphi_l = \left[ f \Delta \sigma \int_a^b \int_c^d K'_u(x', y', x, y, u_l^k(x, y)) dx' dy' \right] \times \left[ f \Delta \sigma \int_a^b \int_c^d K'_u(x, y, x', y', u_l^k(x, y)) dx dy \right],$$

where  $K'_u(x', y', x, y, u_l^k(x, y))$  is transposed kernel function of  $K'_u(x, y, x', y', u_l^k(x, y))$ ,  $A'(u_l^k)^*$  is a transposed derivative operator in  $u_l^k$ . The value  $\varphi_l$  depends on  $u_l^k$ . The process (8) is implemented in discrete form

$$u_{l,i}^{k+1} = u_{l,i}^k - \gamma \frac{1}{\varphi_{l,i}} w_{l,i} \left[ \{A'(u_l^k)^T (A(u^k) - f_\delta)\}_i + \alpha (u_{l,i}^k - u_{l,i}^0) \right], \quad (9)$$

where

$$\varphi_{l,i} = \left[ f \Delta \sigma \sum_{k=1}^N \sum_{m=1}^M K'_u(x'_k, y'_m, \{x, y\}_i, u_{l,i}^k) \Delta x' \Delta y' \right] \times \left[ f \Delta \sigma \sum_{k=1}^N \sum_{m=1}^M K'_u(x_k, y_m, \{x', y'\}_i, u_l^k(x_k, y_m)) \Delta x \Delta y \right],$$



Here we don't need computation of the inverse of matrix  $A'(u^k)^T A'(u^k) + \alpha I$ . It makes this method more economical for numerical solution then (7) which computational complexity is  $O(n^3)$  because of multiplication  $A'(u^k)^T A'(u^k)$  and matrix  $A'(u^k)^T A'(u^k) + \alpha I$  inversion. The computational complexity of (8) is  $O(n^2)$  because the most time-consuming operation here is  $A'(u^k)^T$  matrix elements calculation and matrix-vector multiplication.

Discretizing equation (6) on the  $n = M \times N$  grid with the given right-hand side  $\Delta g(x', y')$  and approximating integral operator  $A(u)$  using the quadrature formula, we obtain the right-hand side  $F(x', y')$  of  $M \times N$  dimension, the solution vector  $u(x, y) = [u_1(x, y), \dots, u_L(x, y)]$  of  $L \times M \times N$  dimension, the derivative matrix of operator  $A'(u^k)$  of  $(M \times N) \times (L \times M \times N)$  dimension, and the system of nonlinear equations

$$A_n[u] = F_n. \quad (10)$$

The  $\|A_n[u^k] - F_n\|/\|F_n\| < \varepsilon$  relative error condition for comparing the exact and numerical solutions with a sufficiently small  $\varepsilon$  is taken as the termination criterion.

### 3. Optimization, parallelization and implementation

A big size matrices in LM algorithm require large amounts of memory. For example, when  $L = 3$ ,  $M = N = 1000$  the matrix  $A'(u^k)^* A'(u^k)$  type of double allocates  $\approx 67$  Tb. Also full matrix-matrix multiplication is very computationally expensive problem. So to reduce memory allocation the decision was made to make all matrix-matrix and matrix-vector computations flying: a matrix element is calculated at the time of access to this element. Let it show.

Previously the system of non-linear equations (10) reduces to the SLAE:

$$B(u^k)u^{k+1} \equiv [A'(u^k)^T A'(u^k) + \alpha I]u^{k+1} = b, \quad (11)$$

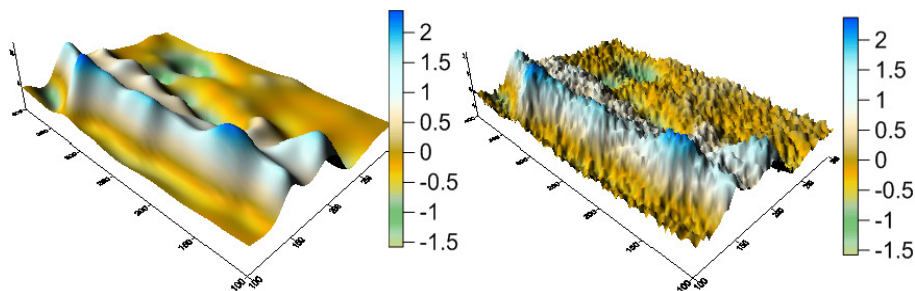
where  $b \equiv [A'(u^k)^T A'(u^k) + \alpha I]u^k - \Lambda A'(u^k)^T (A(u^k) - f_\delta)$ . Here we obtain  $A'(u^k)^T (A(u^k) - f_\delta)$  and  $[A'(u^k)^T A'(u^k)]u^k$  on the fly. Within the "associative law"  $[A'(u^k)^T A'(u^k)]u^k$  equals to  $A'(u^k)^T [A'(u^k)u^k]$ , so "on the fly" technique makes it possible to avoid matrix to matrix multiplication replacing it matrix-vector twice operation. Further the system (11) can be solved by iterative gradient-type methods, minimal residual method e.g [12, 13]. A method chosen in this work is a minimal residual method.

Parallel algorithms for solving (6) are implemented numerically on the multicore Intel Xeon processor and NVIDIA Tesla M2050 graphics processors unit incorporated in the parallel computing system Uran at the Institute of Mathematics and Mechanics of the Ural Branch of RAS. The parallel algorithms are implemented on the multicore Intel Xeon processor using the OpenMP technology and Intel MKL library and on NVIDIA Tesla GPUs using the CUDA technology and CUBLAS library.

For the multicore Intel Xeon processor, the optimization of the vector-matrix operations using the Intel Xeon compiler options and the loop vectorization using the directive `#pragma simd` are implemented.

### 4. Results of numerical experiments

The structural inverse gravimetry problem of finding model interfaces  $S_1, S_2, S_3$  for the four-layer medium with the density jumps was solved using the quasi-model original gravitational data and with uniform noise with an amplitude of 15% noise for the grids  $100 \times 100$  km<sup>2</sup> and  $1000 \times 1000$  km<sup>2</sup>. The gravitational field (fig. 2) is a real but density jumps are taken from model, the model surfaces are based on the quasi-real surfaces constructed in work [9].



**Рис. 2.** Total gravitational field (left) and 15% noised total field (right) (mGal)

The distances to the asymptotic planes were taken as  $H_1 = 8$  km,  $H_2 = 15$  km and  $H_3 = 30$  km. The density jumps were  $\Delta\sigma_1 = 0,2$  g/cm<sup>3</sup>,  $\Delta\sigma_2 = 0,1$  g/cm<sup>3</sup>,  $\Delta\sigma_3 = 0,1$  g/cm<sup>3</sup>. The grid steps were equal to  $\Delta x = 2$  km,  $\Delta y = 3$  km.

The part a) of the fig. 3 shows model interfaces  $S_1, S_2, S_3$ . The part b) shows reconstructed interfaces by LM and the part c) shows CLM results. The parts d) and c) shows reconstructed interfaces by LM and CLM from noised gravitational field.

The table presents the computation times for solving the gravity problem in the three-layer medium for model interfaces with/without noise by the using LM, CLM methods for the grids of  $100 \times 100$  and  $1000 \times 1000$  dimensions. The weight factors were obtained from preliminary selected fields by formula from [7] with parameters  $\alpha = 1, \beta = 1, 1$ . The regularization parameter  $\alpha = 10^{-3}$  and the dumping factor  $\gamma = 1$  were taken for both methods. The termination criterion  $\varepsilon$  was set to 0.25. In the second column of the table number of iterations for gravitational data without noise is written, in the third column number of iterations for gravitational data with 15% noise is shown. The relative errors  $\delta_i = \|u_a - u_e\|/\|u_e\|$  for comparing the exact  $u_e$  and numerical solution  $u_a$  for each  $i$  layer are shown (for original gravitational data). In the last columns the solution times are shown:  $T_1$  is the solution time on one core of Intel Xeon,  $T_2$  is the solution time on eight cores of Intel Xeon,  $T_3$  is the solution time on NVIDIA Tesla M2050 GPU. Data in the top substrings corresponds to  $100 \times 100$  grid and data in the bottom substrings corresponds to  $1000 \times 1000$  grid.

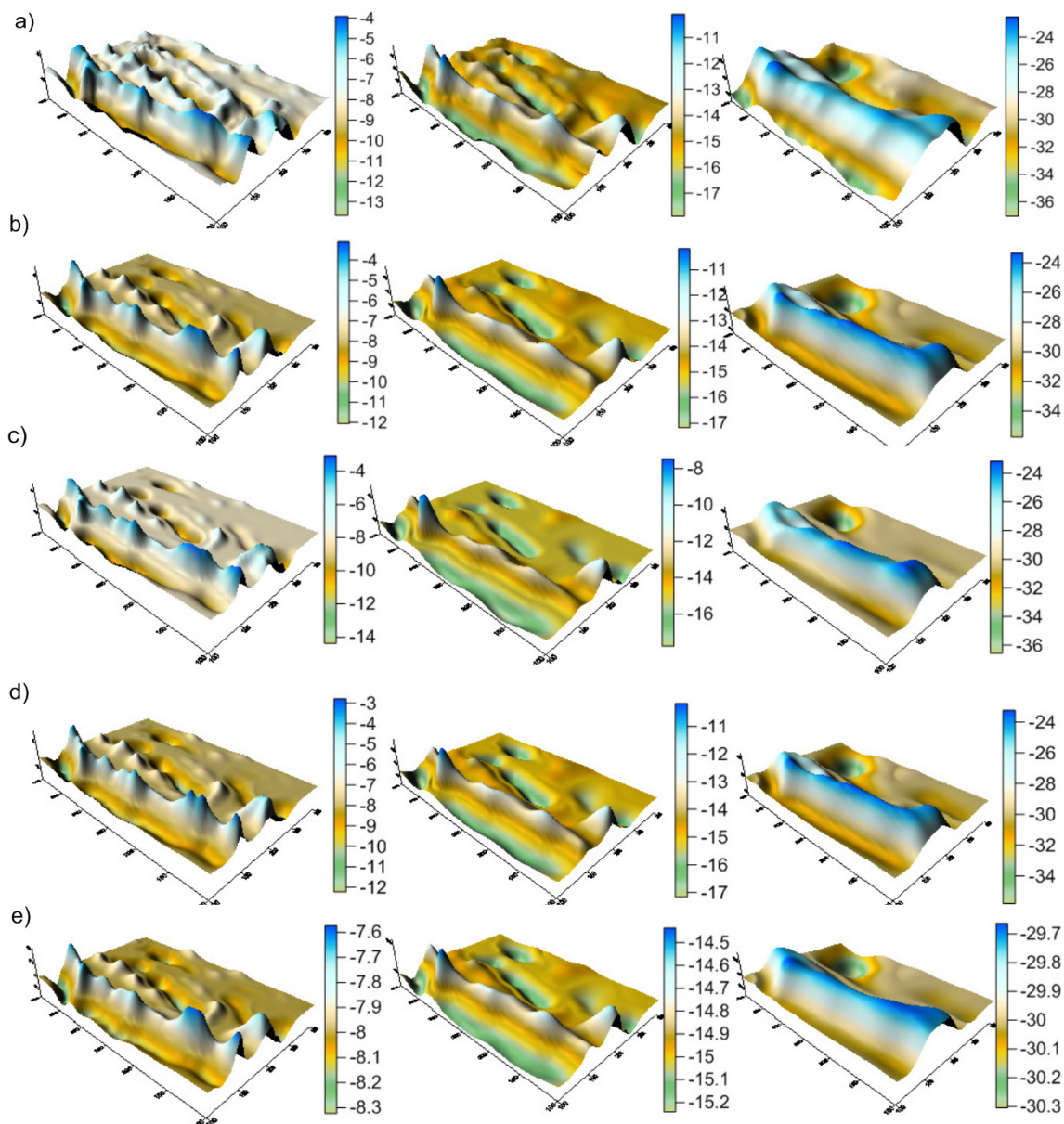
**Table**

Relative errors and computation times

Method	$N_{0\%}$	$N_{15\%}$	$\delta_1$	$\delta_2$	$\delta_3$	$T_1$	$T_2$	$T_3$
LM	30	57	0,052	0,026	0,051	4 min. 6 sec.	2 min. 15 sec.	22 sec.
						11 h. 40 min.	1 h. 25 min.	35 min.
CLM	10	19	0,051	0,035	0,060	33 sec.	16 sec.	2 sec.
						1 h. 12 min.	10 min.	3 min.

## Conclusion

On a base of Levenberg–Marquardt and componentwise Newton type algorithms a Levenberg–Marquardt method is proposed. This method joins advantages of Levenberg–Marquardt scheme in solving gravity multilayer problem and simplicity in Levenberg–Marquardt approach in the Gauss–Newton method. At the same time a regularized Levenberg–Marquardt type method avoids some of the complexities associated with using classic



**Рис. 3.** Comparison of the exact solutions with numerical with parts a) exact solutions, b) numerical LM solutions, c) numerical CLM solutions, d) numerical LM solutions with noised field, e) numerical CLM solutions with noised field

Levenberg–Marquardt method. At first, the inverting an ill-conditioned matrices using internal iterative process. In the second place, matrix-to matrix multiplying entails high computational complexity and big memory consumption. This problem may be solved by "on the fly" technique. The results of numerical experiments show that CLM method has better convergence then classic LM. The both methods are resistant to uniform noise. For large-scale grids, when the data cannot be stored in the memory, "on the fly" technique is the fastest. The computations' acceleration and efficiency on multi-core and graphic accelerators are sufficient. At small grid sizes, the acceleration  $S_n < n$ , where  $n$  is the number of processors, but when the grid size increases it is equalized  $S_n \approx n$  and an efficiency  $E_n \approx 1$ . This means a high resource parallelism of algorithms.

In the future, the question of theoretical interest of the Levenberg–Marquardt method concerns investigating its convergence properties, the conditions on the kernel of the integral operator in equation (1). The obtained conclusions will be useful for another applications.

*Acknowledgments.*

The author expresses his gratitude to his colleague Vladimir Misilov from the Institute of Mathematics and Mechanics of the UrB of RAS and to the researchers team from Institute of Geophysics of the UrB of RAS for providing gravity data used in section 4. Author expresses his deep appreciation to his scientific adviser Elena Akimova for useful comments.

*This work was partly supported by the UrB of the RAS within the framework of the Program of the Presidium RAS (p. 15-7-1-3) and also was partly supported by the RFBR (p. 15-01-00629 A).*

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Akimova E., Skurydina A.A. Componentwise Newton Type Method for Solving the Structural Inverse Gravity Problem. XIV EAGE International Conference — Geoinformatics: Theoretical and Applied Aspects (Kiev, Ukraine, 11–14 May, 2015). DOI: 10.3997/2214-4609.201412361.
2. Akimova E., Skurydina A. On Solving the Three-Dimensional Structural Gravity Problem for the Case of a Multilayered Medium by the Componentwise Levenberg–Marquardt Method. XV EAGE International Conference — Geoinformatics: Theoretical and Applied Aspects (Kiev, Ukraine, 10–13 May, 2016). pp. 181–184. DOI: 10.3997/2214-4609.201600505.
3. Vasin V.V., Perestoronina G.Ya. Levenberg–Marquard Method and Its Modifying Variants for Solutions to Nonlinear Equations with Applications to Inverse Gravimetry Problem. Proceedings of the Institute of Mathematics and Mechanics UrB of RAN. 2011. vol. 11, no. 2. pp. 53–61. DOI: 10.1134/S0081543813020144.
4. Kaltenbacher B., Neubauer A., Scherzer O. Iterative Regularization Methods for Nonlinear Ill-Posed problems. Berlin, New York. Walter de Gruyter, 2008. 194 p.
5. Hanke M. A Regularization Levenberg–Marquardt Scheme, with Applications to Inverse Groundwater Filtration Problems. Inverse Problems. 1997. vol. 13, no. 1. pp. 79–95.
6. Vasin V.V. The Levenberg–Marquardt Method for Approximation of Solutions of Irregular Operator Equations. Automation and Remote Control. 2012. vol. 73, no. 3. pp. 440–449 DOI: 10.1134/S0005117912030034.
7. Akimova E.N., Martyshko P.S., Misilov V.E. Algorithms for Solving the Structural Gravity Problem in a Multilayer Medium. Doklady Earth Sciences. 2013. vol. 453, no. 2. pp. 1278–1281. DOI: 10.1134/S0081543813020144.
8. Martyshko P.S., Prutkin I.L. Technology of Separation of Gravitational Field Sources in Depth. *Geofizicheskii Zhurnal* [Geophysical Journal]. 2003. vol. 25, no. 3. pp. 159–168. (in Russian)

9. Martyshko P.S., Akimova E.N., Misilov V.E. Solving the Structural Inverse Gravity Problem by the Modified Gradient Methods // *Izvestiya, Physics of the Solid Earth*. 2016. vol. 52, no. 5. pp. 704–708. DOI: 10.1134/S1069351316050098.
10. Akimova E.N., Martyshko P.S., Misilov V.E. A Fast Parallel Gradient Algorithm for Solving Structural Inverse Gravity Problem. XIII International Conference on Numerical Analysis and Applied Mathematics: AIP Conference Proceedings (Rodos, Greece, 22–28 September, 2015). 1648, 850063. DOI: 10.1063/1.4913118.
11. Akimova E.N., Misilov V.E. A Fast Componentwise Gradient Method for Solving Structural Inverse Gravity Problem. International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management (SGEM): Proceedings of 15th Intern. Multidisciplinary Scientific GeoConference SGEM. (Albena, Bulgaria, 18–24 June, 2015). vol. 3. pp. 775–782.
12. Vasin V.V., Eremin I.I. Operators and Iterative Processes of Fejer Type. Theory and Applications. Walter de Gruyter, Berlin. 2009. 155 p. DOI: 10.1515/9783110218190.
13. Bakhvalov N.S., Zhidkov N.P., Kobel'kov, G.M. Numerical Methods. Nauka, Moscow. 1987. 600 p.

УДК 004.853

DOI: 10.14529/cmse170301

## РЕГУЛЯРИЗОВАННЫЙ ПОКОМПОНЕНТНЫЙ МЕТОД ТИПА ЛЕВЕНБЕРГА—МАРКВАРДТА В ПРИЛОЖЕНИИ К ОБРАТНОЙ ЗАДАЧЕ ГРАВИМЕТРИИ В МНОГОСЛОЙНОЙ СРЕДЕ И ЕГО ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ\*

© 2017 А.Ф. Скурыдина

*Институт математики и механики им. Н. Н. Красовского УрО РАН  
(620049 Екатеринбург, ул. им. Софьи Ковалевской, д. 16),*

*Уральский федеральный университет имени первого Президента России Б. Н. Ельцина  
(620002 Екатеринбург, ул. Мира, д. 19)*

*E-mail: afinapal@gmail.com*

Поступила в редакцию: 05.06.2017

Структурная обратная задача гравиметрии в многослойной среде является важнейшей из геофизических задач. До недавнего времени задача сводилась к разделению гравитационных полей и восстановлению неизвестных слоев по отдельности, сейчас актуальны методы, которые позволяют находить неизвестные поверхности одновременно. Для решения интегрального уравнения Урысона первого рода, описывающего данную задачу, предложены и исследуются регуляризованные методы на основе метода Левенберга—Марквардта с использованием весовых множителей. Предложен новый метод покомпонентного типа на основе схемы Левенберга—Марквардта. Сравнивается регуляризованный покомпонентный метод типа Левенберга—Марквардта с классическим. Для классического метода Левенберга—Марквардта предложены некоторые вычислительные оптимизации. Численные эксперименты на примере модельных гравитационных данных позволяют сравнить скорость сходимости, относительные ошибки и времена

\* Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2017».

выполнения программ классического метода Левенберга–Марквардта и покомпонентного. Параллельные программы, реализующие данные алгоритмы, разработаны с использованием технологий CUDA и OpenMP.

*Ключевые слова:* регуляризация по Тихонову, регуляризованный метод Левенберга–Марквардта, регуляризованный покомпонентный метод типа Левенберга–Марквардта, обратная задача гравиметрии в многослойной среде.

## ОБРАЗЕЦ ЦИТИРОВАНИЯ

Skurydina A.F. A Regularized Levenberg–Marquardt Type Method Applied to the Structural Inverse Gravity Problem in a Multilayer Medium and its Parallel Realization // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 5–15. DOI: 10.14529/cmse170301.

## Литература

1. Akimova E., Skurydina A. A Componentwise Newton Type Method for Solving the Structural Inverse Gravity Problem // XIV EAGE International Conference — Geoinformatics: Theoretical and Applied Aspects (Kiev, Ukraine, 11–14 May, 2015). DOI: 10.3997/2214-4609.201412361.
2. Akimova E., Skurydina A. On Solving the Three-Dimensional Structural Gravity Problem for the Case of a Multilayered Medium by the Componentwise Levenberg–Marquardt Method // XV EAGE International Conference — Geoinformatics: Theoretical and Applied Aspects (Kiev, Ukraine, 10–13 May, 2016). P. 181–184. DOI: 10.3997/2214-4609.201600505.
3. Васин В.В., Пересторонина Г.Я. Метод Левенберга–Марквардта и его модифицированные варианты для решения нелинейных уравнений с приложением к обратной задаче гравиметрии // Труды Института математики и механики УрО РАН. 2011. Т. 11 № 2, С. 53–61. DOI: 10.1134/S0081543813020144.
4. Kaltenbacher B., Neubauer A., Scherzer O. Iterative Regularization Methods for Nonlinear Ill-Posed Problems. Berlin, New York. Walter de Gruyter, 2008. 194 p.
5. Hanke M. A Regularization Levenberg–Marquardt Scheme, with Applications to Inverse Groundwater Filtration Problems // Inverse Problems. 1997. Vol. 13, No. 1. P. 79–95.
6. Васин В.В. Метод Левенберга–Марквардта для аппроксимации решений нерегулярных операторных уравнений // Автоматика и телемеханика. 2012. Т. 73, № 3. С. 28–38. DOI: 10.1134/S0005117912030034.
7. Акимова Е.Н., Мартышко П.С., Мисиллов В.Е. Методы решения структурной задачи гравиметрии в многослойной среде // Доклады Академии наук. 2013. Т. 453, № 2. С. 1278–1281. DOI: 10.1134/S0081543813020144.
8. Мартышко П.С., Пруткин И.Л. Технология разделения источников гравитационного поля на глубине // Геофизический журнал. 2003. Т. 25, № 3. С. 159–168.
9. Martyshko P.S., Akimova E. N., Misilov V. E. Solving the Structural Inverse Gravity Problem by the Modified Gradient Methods // Izvestiya, Physics of the Solid Earth. 2016. Vol. 52, No. 5. P. 704–708. DOI: 10.1134/S1069351316050098.
10. Akimova E.N., Martyshko P.S., Misilov V.E. A Fast Parallel Gradient Algorithm for Solving Structural Inverse Gravity Problem // XIII International Conference on Numerical Analysis and Applied Mathematics: AIP Conference Proceedings (Rodos, Greece, 22–28 September, 2015). 1648, 850063. DOI: 10.1063/1.4913118.

11. Akimova E.N., Misilov V.E. A Fast Componentwise Gradient Method for Solving Structural Inverse Gravity Problem. International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management (SGEM) // Proceedings of 15th Intern. Multidisciplinary Scientific GeoConference SGEM. (Albena, Bulgaria, 18–24 June, 2015). Vol. 3. pp. 775–782.
12. Vasin V. V., Eremin I. I. Operators and Iterative Processes of Fejer Type. Theory and Applications. Walter de Gruyter, Berlin. 2009. 155 p. DOI: 10.1515/9783110218190.
13. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. - Численные методы. Москва: Наука, 1987. 600 с.

Скурыдина Алия Фиргатовна, м.н.с., отдел некорректных задач анализа и приложений Института математики и механики им. Н.Н. Красовского УрО РАН; ассистент, департамент радиоэлектроники и связи Уральского Федерального университета им. первого Президента России Б. Н. Ельцина (Екатеринбург, Российская Федерация)

## PREIMAGE ATTACK ON MD4 HASH FUNCTION AS A PROBLEM OF PARALLEL SAT-BASED CRYPTANALYSIS\*

© 2017 I.A. Gribanova, O.S. Zaikin, I.V. Otpuschennikov,  
A.A. Semenov

*Matrosov Institute for System Dynamics and Control Theory SB RAS  
(Lermontova st. 134, Irkutsk, 664033, Russia)*

*E-mail: the42dimension@gmail.com, zaikin.icc@gmail.com, otilya@yandex.ru,  
biclop.rambler@yandex.ru*

Received: 04.05.2017

In this paper we study the inversion problem of MD4 cryptographic hash function developed by R. Rivest in 1990. By MD4-k we denote a truncated variant of MD4 hash function in which k represents a number of steps used to calculate a hash value (the full version of MD4 function corresponds to MD4-48). H. Dobbertin has showed that MD4-32 hash function is not one-way, namely, it can be inverted for the given image of a random input. He suggested to add special conditions to the equations that describe the computation of concrete steps (chaining variables) of the considered hash function. These additional conditions allowed to solve the inversion problem of MD4-32 within a reasonable time by solving corresponding system of equations. The main result of the present paper is an automatic derivation of “Dobbertin’s conditions” using parallel SAT solving algorithms. We also managed to solve several inversion problems of functions of the kind MD4-k (for k from 31 up to 39 inclusive). Our method significantly outperforms previously existing approaches to solving these problems.

*Keywords: cryptanalysis, hash function, inversion problem, MD4, SAT, parallel computing, MPI.*

### FOR CITATION

Gribanova I.A., Zaikin O.S., Otpuschennikov I.V., Semenov A.A. Preimage Attack on MD4 Hash Function as a Problem of Parallel Sat-Based Cryptanalysis. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 16–27. DOI: 10.14529/cmse170302.

### Introduction

Hash function is a function which maps binary words of arbitrary length into binary words of fixed length. More precisely, a hash function is a total computable discrete function of the kind

$$\chi : \{0, 1\}^* \rightarrow \{0, 1\}^C, C = const. \quad (1)$$

By  $\{0, 1\}^k, k \in \mathbb{N}$  we denote the set formed by all  $2^k$  different vectors of length  $k$ . By  $\{0, 1\}^*$  we denote the set of all binary words of an arbitrary finite length.

Hash functions are used in various areas of computer science, for example, to speed up an access to large data sets. In cryptography and information security the range of issues that can be solved using hash functions is especially wide. Cryptographic hash functions meet additional requirement: corresponding functions should be hard to invert. In particular, not only the inversion problem (i.e. the problem of finding a preimage for a given hash value) but also the problem of finding collisions (i.e. the problem of finding an arbitrary pair of messages that give the same hash) should be hard.

\*The paper is recommended for publication by the Program Committee of the International Scientific Conference “Parallel Computational Technologies (PCT) 2017”.



In [1, 2] a method of finding collisions for hash functions from the MD family, widely used at that time, was presented. Further this problem was discussed in a number of papers, wherein various methods for constructing collisions were employed. The possibility of effective collision search for the MD family hash functions resulted in their exclusion from wide usage and replacement by other hash functions (mainly by the SHA family hash functions) in most cryptographic systems. However, today even for MD4 (the weakest hash function from the MD family) there are no successful results in solving the problem of its inversion. Moreover, there are no algorithms that would appear to be much more computationally stronger than the method proposed in [3] (one of the first papers in which the problem of MD4 inversion was studied).

The main idea of the attack proposed in [3] consists in considering the truncated variants of MD4 (with less than 48 steps of the hash value calculation) and some additional conditions to the equations, defining the corresponding function. In some cases this approach allows to solve the corresponding system of equations on a parallel computing system in reasonable time.

In the present paper we study the MD4 inversion problems using parallel algorithms for solving Boolean satisfiability problem (SAT). This paper is organized as follows. In Section 1 we present the necessary information concerning the algorithmic features of MD4 and the basic idea of H. Dobbertin's attack. In Section 2 we describe the foundations of the SAT-based cryptanalysis. We also consider the reduction of the MD4 inversion problem to SAT and give a brief description of parallel algorithms applied to the obtained SAT instances. Section 3 presents the results of computational experiments and comparison with the results obtained in previous works.

## 1. Structure of MD4 and Basic Idea of Dobbertin's Attack

The cryptographic hash function MD4 [4] was developed by R. Rivest in 1990. This function is one of the first examples of practical implementation of the Merkle-Damgard construction [5, 6]. The basic paradigm of the Merkle-Damgrad construction consists in the fact that a hash value is a result of a sequence of similar actions, which is written into a special register. Further this register is called a hash register. At the initial step the hash register is filled with some known value (Initial Value, IV). In case of MD4

$$IV = \{0x67452310, 0xEFCDAB89, 0x98BADCFE, 0x10325476\}. \quad (2)$$

At the next steps the hash register states are modified by mixing the current state with parts of the input message (which should be hashed). Like many other hash functions, MD4 works with input message divided into blocks with 512 bits in each. The value of MD4 is a binary 128-bit word. Basic primitives of MD4 are 32-bit words (as in many hash functions which were developed later): 512-bit block is divided into 16 32-bit words, 128-bit hash is divided into 4 32-bit words.

At the initial stage MD4 algorithm uses a special padding procedure. After that MD4 compression function  $f_{MD4}$  is applied to the obtained 512-bit block. The result of  $f_{MD4}$  is 128-bit hash of the considered block. Hereinafter we don't take padding procedure into account. Thus, we consider only the inversion problem of  $f_{MD4}$ . The process of calculation of this function is an iterative procedure which is divided into 48 steps. On each step the value of one hash register cell (filled with 32-bit word) is updated. Identifiers of these cells are called chaining variables. So each chaining variable takes values from  $\{0, 1\}^{32}$ . We denote the chaining variables by letters  $a, b, c, d$ . The process of hash calculation is divided into 3 rounds, each of them consists of 16 steps. In each round a certain round function is used, which operates with 32-bit variables. The calculation scheme of  $f_{MD4}$  is shown in Fig.

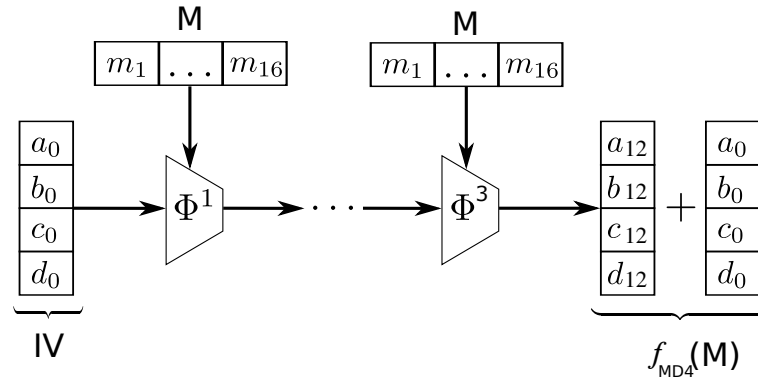


Fig. The calculation scheme of  $f_{MD4}$

Let us briefly comment on Fig. By  $M = m_1 | \dots | m_{16}$  we denote the 512-bit input block. By  $\Phi^1 - \Phi^3$  the round transformations are denoted. For example, for the first 4 steps the corresponding transformations of chaining variables are as follows:

$$\begin{aligned}
 a_1 &= (a_0 + \phi^1(b_0, c_0, d_0) + m_1 + t_1) \lll s_a^1, \\
 d_1 &= (d_0 + \phi^1(a_1, b_0, c_0) + m_2 + t_1) \lll s_d^1, \\
 c_1 &= (c_0 + \phi^1(d_1, a_1, b_0) + m_3 + t_1) \lll s_c^1, \\
 b_1 &= (b_0 + \phi^1(c_1, d_1, a_1) + m_4 + t_1) \lll s_b^1.
 \end{aligned} \tag{3}$$

In these formulas “+” stands for  $\text{mod } 2^{32}$  summation of the corresponding numbers, “ $\lll s$ ” stands for the cyclic shift of a 32-bit word to  $s$  positions to the left. Constants  $t_i$  and  $s_{a,b,c,d}^i, i \in \{1, 2, 3\}$  (here  $i$  is the round number) are known from the specification of the algorithm (for example,  $t_1 = 0, t_2 = 0x5A827999, t_3 = 0x6ED9EBA1$ ).

In each round with number  $i, i \in \{1, 2, 3\}$  all chaining variable are updated 4 times by applying the round function  $\phi^i$ . The MD4 round functions are as follows:

$$\begin{aligned}
 \phi^1(X, Y, Z) &= (X \wedge Y) \vee (\neg X \wedge Z), \\
 \phi^2(X, Y, Z) &= (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z), \\
 \phi^3(X, Y, Z) &= X \oplus Y \oplus Z.
 \end{aligned} \tag{4}$$

Arguments of these functions are 32-bit words. All logical operations are performed component-wise over the corresponding vectors. The summation of the  $IV$  vector and the value of hash register at the end of the third round presented in Fig. is called a finalization stage.

By MD4- $k$  we denote the hash function which corresponds to the execution of  $k$ -steps of MD4 algorithm applied to  $IV$  with finalization stage performed after these steps. For example, MD4-48 corresponds to the full-round version of MD4 hash function. An arbitrary function of the kind MD4- $k, k \in \{1, \dots, 48\}$  is called the truncated version of MD4.

In [3] an algorithm aimed at inversion of MD4-32 using an ordinary PC was proposed. As a result, it was shown that 2-round version of MD4 hash function is not one-way. The basic idea of the attack is as follows. By analyzing the first two round functions it can be concluded that assignment of some chaining variables with some constant leads to finding the majority of words from the set  $\{m_1, \dots, m_{16}\}$  in a short time.

In particular, H. Dobbertin suggested to fix chaining variables values with some constant  $K$  at the steps with numbers 13, 17, 21, 25, 14, 18, 22, 26, 15, 19, 23, 27. Then the value of the chaining variable  $b$  at the 28-th step is varied. It should be noted that in this attack the value of variable  $b$  is recalculated at the 28-th step, but it isn't changed for the other 3 steps. Thus,

if some value of  $b$  is determined, then the same value of this variable is also determined for the steps 29, 30, 31. The fixation of  $K$  and  $b$  at the steps mentioned above gives us an opportunity to derive the values of variables  $m_1, m_2, \dots, m_{12}$  and  $m_{16}$  of the input message. Their usage makes it possible to derive the values of several more chaining variables and, finally, the value of  $b$  at the 28-th step. The latter value can differ from the one fixed previously. In this case the attempt is considered to be unsuccessful and the sequence of actions described above is repeated for new  $b$  (in this case constant  $K$  may remain unchanged).

H. Dobbertin mentioned that in order to achieve successful results in the described attack one needs to make about  $2^{32}$  attempts of selection of the value of  $b$  on the 28-th step. So the proper implementation of the presented algorithm can make it possible to achieve the successful result even on a weak PC.

## 2. Reduction of the MD4-k Inversion Problems to SAT and the SAT Version of Dobbertin's Attack

In this section we describe some new techniques for SAT-based cryptanalysis and apply them to the inversion problems of MD4 hash function.

SAT-based cryptanalysis is a relatively new direction in cryptanalysis implying the usage of algorithms for solving Boolean satisfiability problem (SAT) for the inversion of cryptographic functions. Let us remind that for SAT it is necessary for an arbitrary Boolean formula  $F$  to decide whether it is satisfiable or not, i.e. if there exists an assignment of Boolean variables from this formula that makes it TRUE. Using Tseitin transformations [7] the Boolean satisfiability problem for  $F$  can be reduced to SAT in the Conjunctive Normal Form (CNF) in polynomial time on the size of  $F$  description. Hereinafter by SAT we mean the problem of satisfiability of an arbitrary CNF.

According to the Cook theorem, a wide class of combinatorial problems can be effectively reduced to SAT, including the inversion problems of cryptographic functions: for a given image from a range of values of considered function to find a preimage from its domain (assuming that function is defined by known algorithm). Today there is a number of automatic translation systems designed for effectively construction of SAT encodings for inversion of cryptographic functions [8–11]. In all our computational experiments we use the Transalg system [11].

Once a SAT encoding for the inversion problem of the considered cryptographic function is built, the corresponding SAT instance is ready to be solved. A variety of algorithms can be used for this purpose. However, according to numerous computational experiments, CDCL-based SAT solvers [12] are better suited for inversion of cryptographic functions. The survey of algorithms and technologies underlying modern CDCL-solvers can be found in [13].

It should be noted that cryptanalysis problems in the form of SAT are usually extremely hard even for the best-known SAT-solvers (except for the inversion problems of some weak functions, e.g. the Geffe generator). Thus, for functions with serious cryptographic resistance additional considerations should be used (these considerations may arise from the algorithmic features of these functions). As usual, it is hard to avoid using parallel computations for the real attacks. Below we provide a brief description of the parallel SAT technologies, which we applied to the inversion problems of MD4-k hash functions.

Today there are two main approaches to the parallel solving of SAT instances: the portfolio approach and the partitioning approach [14]. Portfolio approach can be considered as a multi-threaded parallelism, while partitioning-approach is the large-block data parallelism.

According to the portfolio approach, multiple copies of a SAT solver are used and, roughly speaking, each of these copies goes through the common search space in different directions sharing accumulated information. In the partitioning approach a search space is divided into disjoint subdomains which are processed separately. Thus, the partitioning approach is better suited to solve hard SAT instances in distributed computing systems, while the portfolio approach shows the best results in multithreaded systems.

As already mentioned, apart from parallelism, it is often possible to accelerate the inversion of cryptographic functions by taking into account various features of these functions. For example, in order to achieve results in solving the SAT instances for finding collisions from MD family hash functions it is necessary to add special conditions called differential paths on chaining variables to the corresponding SAT-encodings. The first successful attack of this kind has been shown in [15], where the authors have added to the SAT encodings conditions specifying the differential paths described in [1, 2].

A similar situation occurs with respect to the inversion problems of MD4-k functions. Dobbertin's conditions described above represent additional constraints which significantly reduce the search space. The following questions are: how successful these additional constraints are, and are there more effective ones? The answers to these questions are discussed in the present paper.

Before proceeding to the submission of our results, let us note that SAT solvers can be considered as a means of "intellectual search". Consequently, the use of a SAT solver must be aimed primarily at the automation of a large number of similar operations. The technique of information preservation and non-chronological backtracking used in CDCL-solvers can provide essential reduction of the amount of calculation in comparison with the exhaustive search. Similar arguments were given in [15], where the authors emphasize that they have used the SAT approach primarily in order to automate the message modification phase forming the main part of the attack described in [1, 2].

In the Dobbertin's attack the search through all possible values of variable  $b$  corresponding to the 28-th step of the hash function is performed, wherein the value of  $K$  may be fixed for different values of  $b$ . In the attack proposed below we use the value  $K = 0$  (as in [16]) and consider the inversion problem of  $1^{128}$  hash, i.e. assuming that the hash value consists of 128 ones.

The main difference between our attack and Dobbertin's attack consists in the fact that we search through all possible variants of assignment of chaining variables with constant  $K = 0$  using a parallel SAT solver. We do not use assignment of the variable  $b$  on the 28-th step assuming that SAT solver automatically adjusts the values of the unknown variables for a particular combination of chaining variables assigned with constant  $K = 0$ .

For the purpose of the automatic search through different combinations of chaining variables we use additional variables called switching variables. Let us describe the corresponding technique. By  $C^* = C(f_{MD4-k}(M) = 1^{128})$  we denote CNF encoding of the inversion problem of  $f_{MD4-k}$  function in point  $1^{128}$ . As in Dobbertin's attack first we consider the inversion problem of  $f_{MD4-32}$  function. In fact we need a procedure that will allow to quickly add to the  $C^*$  various combinations of chaining variables on the steps with numbers from 1 to 32 assigned with constant  $K = 0$ . By  $z^j$  we denote a 32-bit chaining variable of  $j$ -step. For  $z^j$  there are 32 corresponding Boolean variables  $y_1^j, \dots, y_{32}^j$ . Let us consider the elementary conjunction:

$$R^j = \neg y_1^j \wedge \dots \wedge \neg y_{32}^j. \tag{5}$$

Obviously,  $R^j$  is equal to 1 if and only if variable  $z^j$  is assigned with constant  $K = 0$ . For each  $j \in \{1, \dots, 32\}$  let us consider a new Boolean variable  $s^j$ , which we associate with  $R^j$  by the following formula:

$$F^j = (\neg s^j \vee \neg y_1^j) \wedge \dots \wedge (\neg s^j \vee \neg y_{32}^j). \quad (6)$$

Obviously, the substitution of  $s^j = 1$  into  $F^j$  gives a formula which is logically equivalent to  $R^j$ . The substitution of  $s^j = 0$  into  $F^j$  gives a constant of 1. We shall consider the following CNF:

$$\tilde{C} = C^* \wedge F^1 \wedge \dots \wedge F^{32}. \quad (7)$$

According to the mentioned above, the substitution of any assignment  $s \in \{0, 1\}^{32}$  of switching variables:

$$s^j = \alpha^j, \alpha^j \in \{0, 1\}, j \in \{1, \dots, 32\}, \quad (8)$$

in  $\tilde{C}$  gives the inversion problem of MD4-32 function in point  $1^{128}$  with additional assignment with constant  $K = 0$  of chaining variable  $z^j$  for which corresponding  $\alpha^j$  takes value 1. Let us note that such assignment can be considered in application to the problem of the satisfiability of  $\tilde{C}$  as an *assumption* and it is possible to apply incremental SAT technique [17] which allows to store and use the information obtained during processing of various assumptions.

### 3. Computational Experiments

In our experiments we used two SAT solvers. The first one is our parallel SAT solver PDSAT [18], which is based on the partitioning approach. This solver was designed especially for solving SAT instances that encode inversion problems of cryptographic functions. PDSAT is an MPI-program, in which there is one leader process, all the other are computing processes (each process corresponds to 1 CPU core). PDSAT works in two modes. In the estimation mode it searches for a decomposition set with good time estimations. A decomposition set is in fact a set of Boolean variables, for which all their possible assignments are generated. As a result, a family of SAT instances, where each of instances is simpler than the original problem, was obtained. In this mode we use the Monte Carlo approach and various optimization metaheuristics (simulated annealing, tabu search, etc.). In the solving mode PDSAT solves all SAT instances from a family obtained from a given decomposition set. CRYPTOMINISAT [10] is the second SAT solver which was used in our experiments. It is not designed for launching on an MPI cluster, so we launched it on a PC as a sequential program.

At the first stage we considered the MD4-31 inversion problem in point  $1^{128}$ . We launched PDSAT in the solving mode on the decomposition set which consisted of 27 switching variables (see the previous section). Time limit of 0.01 second for each subproblem was used. PDSAT was launched on 5 nodes of the “Academician V.M. Matrosov” computing cluster of Irkutsk supercomputing center SB RAS<sup>1</sup>. Each node of this cluster consists of 2 16-core CPUs AMD Opteron 6276, so 160 CPU cores were used in total. All  $2^{27}$  SAT instances were solved in 10 minutes 21 seconds, the processing of 4.88 % SAT instances was interrupted by time limit. As a result, 6 satisfying assignments (6 solutions of the considered inversion problem) were found. It should be noted that one of the assignments of switching variables with constant  $K = 0$  was in fact the Dobbertin’s conditions. Thus, we found these conditions in automatic mode.

At the second stage we considered the MD4-39 inversion problem, taking into account results from [16], where the authors involved 11 of the 12 Dobbertin’s conditions. We constructed 3

<sup>1</sup><http://www.hpc.icc.ru>

CNFs, in which we fixed the values of 11 switching variables corresponding to aforementioned 11 conditions, and assigned a different number of variables encoding the chaining variable corresponding to the omitted Dobbertin’s condition. The value of this chaining variable was taken from one of the satisfying assignments of MD4-31 inversion problem. In the first CNF the first 8 bits of this chaining variable were assigned. In the second and third CNFs the first 16 and 32 bits of this chaining variable were assigned (respectively).

We launched CRYPTOMINISAT on a PC equipped with the i5-2410M CPU (1 core was used). The best result was obtained on the first CNF – a satisfying assignment was found in 15 minutes. We also employed PDSAT in the solving mode. The best result was obtained on the second CNF – the corresponding problem was solved in 5 seconds on 5 computing nodes (160 CPU cores in total). Thus, if we recalculate this time on the case of a sequential launch, we obtain the time of about 13 minutes. It should be noted, that in [16] this inversion problem was solved in about 8 hours on 1 CPU core. So our approach allowed to solve this problem much faster. In Tab. we show 4 input messages  $M$  found in our experiments by PDSAT and CRYPTOMINISAT. All these messages correspond to the hash value  $1^{128}$ .

Table

Messages for the MD4-39 inversion problem in point  $1^{128}$

No.	Message
1	0xc7c08b1c,0xa57d8667,0xa57d8667,0x07e14fec, 0xa57d8667,0xa57d8667,0xa57d8667,0xa8cea698, 0xa57d8667,0xa57d8667,0xa57d8667,0x28e987ac, 0x4665c5f3,0x8c49173f,0xabc74a06,0x0cd9d788
2	0x40b2a2ff,0xa57d8667,0xa57d8667,0x2b010cef, 0xa57d8667,0xa57d8667,0xa57d8667,0xccfef2c3, 0xa57d8667,0xa57d8667,0xa57d8667,0xf041ded3, 0x5443c70c,0xaadd4c2b,0xe587e70e,0xe5bad382
3	0x40b26b1f,0xa57d8667,0xa57d8667,0xa87153ec, 0xa57d8667,0xa57d8667,0xa57d8667,0xbaee84bb, 0xa57d8667,0xa57d8667,0xa57d8667,0x6a228d63, 0x5960f23b,0x1915d72b,0xf2d3b064,0x7d85d6db
4	0xec3a2319,0xa57d8667,0xa57d8667,0xf87f9cee, 0xa57d8667,0xa57d8667,0xa57d8667,0x80a0aeb0, 0xa57d8667,0xa57d8667,0xa57d8667,0x51c2c922, 0x964933fa,0x0545c48c,0x98968391,0x783c0174

#### 4. Related Work

As we mentioned above, the first successful example of the SAT approach application to the inversion of relevant cryptographic functions was presented in [15]. In that paper a SAT-based variant of attack by X. Wang et al. [1, 2] was suggested. The method used in [15] allows to find one-block collisions for MD4 relatively fast. To find two-block collisions for MD5 much more computational resources must be used. We significantly increased the efficiency of the approach proposed in [15]: the application of SAT encodings constructed by the Transalg system [11] allowed us to generate one-block collisions for MD4 approximately 1000 times faster. By applying

modern multithreaded SAT solvers we managed to find two-block collisions for MD5 in reasonable time using one computing cluster node. As a result, a family of such collisions of special kind (with a large amount of most significant bits) was constructed [19].

The first SAT-based version of Dobbertin's attack was proposed in [16]. The corresponding chaining variables were assigned with constant  $K = 0$ . It should be noted that in that paper the Dobbertin's conditions were used in their original form – with no attempts to justify or derive them. Also in [16] one Dobbertin's condition was thrown out without any justification too.

In [16] it took about 8 hours on the MINISAT solver to invert MD4-39 with the Dobbertin's conditions for chaining variables of the first two rounds. Thus, we can conclude that the effectiveness of the method we suggested is significantly higher than the one proposed in [16].

## Conclusions and Future Work

In this paper, we managed to automatically synthesize conditions from Dobbertin's attack using parallel SAT solving algorithms. We also studied the inversion problem of the MD4-39 hash function. By applying parallel SAT algorithms this problem was solved faster than it was done in previous works. In the nearest future we plan to apply parallel SAT algorithms to inversion problems of some other hash functions (from the MD and SHA hash families).

*This research was funded by Russian Science Foundation (project No. 16-11-10046). Oleg Zaikin and Ilya Otpuschennikov are partially funded by Council for Grants of the President of the Russian Federation (stipends SP-1184.2015.5 and SP-4751.2016.5 respectively).*

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Wang X., Lai X., Feng D., Chen H., Yu X. Cryptanalysis of the Hash Functions MD4 and RIPEMD. *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques. EUROCRYPT'05*. Berlin, Heidelberg: Springer-Verlag, 2005. pp. 1–18. DOI: 10.1007/11426639\_1.
2. Wang X., Yu H. How to Break MD5 and Other Hash Functions. *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques. EUROCRYPT'05*. Berlin, Heidelberg: Springer-Verlag, 2005. pp. 19–35. DOI: 10.1007/11426639\_2.
3. Dobbertin H. The First Two Rounds of MD4 are Not One-Way. *Fast Software Encryption / Ed. by Serge Vaudenay. Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1998. vol. 1372. pp. 284–292. DOI: 10.1007/3-540-69710-1\_19.
4. Rivest R.L. The MD4 Message Digest Algorithm. *Advances in Cryptology - CRYPTO'90, Proceedings / Ed. by Alfred Menezes, Scott A. Vanstone. Lecture Notes in Computer Science*. Springer, 1990. vol. 537. pp. 303–311. DOI: 10.1007/3-540-38424-3\_22.
5. Damgård I.B. A Design Principle for Hash Functions. *Proceedings on Advances in Cryptology. CRYPTO '89*. New York, NY, USA: Springer-Verlag New York, Inc., 1989. pp. 416–427. DOI: 10.1007/0-387-34805-0\_39.

6. Merkle R.C. A Certified Digital Signature. *Proceedings on Advances in Cryptology. CRYPTO '89*. New York, NY, USA: Springer-Verlag New York, Inc., 1989. pp. 218–238. DOI: 10.1007/0-387-34805-0\_21.
7. Tseitin G.S. On the Complexity of Derivation in Propositional Calculus. *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983. pp. 466–483. DOI: 10.1007/978-3-642-81955-1\_28.
8. Erkök L., Matthews J. High assurance programming in Cryptol. *Fifth Cyber Security and Information Intelligence Research Workshop, CSIIRW'09, Knoxville, TN, USA, April 13-15, 2009 / Ed. by Frederick T. Sheldon, Greg Peterson, Axel W. Krings [et al.]*. ACM, 2009. pp. 60. DOI: 10.1145/1558607.1558676.
9. Janjicic P. URSA: a System for Uniform Reduction to SAT. *Logical Methods in Computer Science*. 2012. vol. 8, no. 3. pp. 1–39. DOI: 10.2168/lmcs-8(3:30)2012.
10. Soos M., Nohl K., Castelluccia C. Extending SAT Solvers to Cryptographic Problems . *SAT / Ed. by Oliver Kullmann. Lecture Notes in Computer Science*. Springer, 2009. vol. 5584. pp. 244–257. DOI: 10.1007/978-3-642-02777-2\_24.
11. Otpuschennikov I., Semenov A., Gribanova I., Zaikin O., Kochemazov S. Encoding Cryptographic Functions to SAT Using TRANSALG System. *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016) / Ed. by Gal A. Kaminka, Maria Fox, Paolo Bouquet [et al.]*. *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2016. vol. 285. pp. 1594–1595.
12. Marques-Silva J.P., Sakallah K.A. GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Trans. Computers*. 1999. vol. 48, no. 5. pp. 506–521. DOI: 10.1109/12.769433.
13. Marques-Silva J.P., Lynce I., Malik S. Conflict-Driven Clause Learning SAT Solvers. *Handbook of Satisfiability / Ed. by Armin Biere, Marijn Heule, Hans van Maaren, Toby Walsh*. *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009. vol. 185. pp. 131–153.
14. Hyvärinen A.E.J. Grid Based Propositional Satisfiability Solving. Ph.D. thesis, Aalto University, 2011.
15. Mironov I., Zhang L. Applications of SAT Solvers to Cryptanalysis of Hash Functions. *SAT / Ed. by Armin Biere, Carla P. Gomes. Lecture Notes in Computer Science*. Springer, 2006. vol. 4121. pp. 102–115. DOI: 10.1007/11814948\_13.
16. De D., Kumarasubramanian A., Venkatesan R. Inversion Attacks on Secure Hash Functions Using SAT Solvers. *Theory and Applications of Satisfiability Testing - SAT 2007, Proceedings / Ed. by João Marques-Silva, Karem A. Sakallah. Lecture Notes in Computer Science*. Springer, 2007. vol. 4501. pp. 377–382. DOI: 10.1007/978-3-540-72788-0\_36.
17. Eén N., Sörensson N. Temporal induction by incremental SAT solving. *Electr. Notes Theor. Comput. Sci.* 2003. vol. 89, no. 4. pp. 543–560. DOI: 10.1016/s1571-0661(05)82542-3.
18. Semenov A., Zaikin O. Algorithm for finding partitionings of hard variants of boolean satisfiability problem with application to inversion of some cryptographic functions. *SpringerPlus*. 2016. vol. 5, no. 1. pp. 1–16. DOI:10.1186/s40064-016-2187-4.



19. Bogachkova (Gribanova) I., Zaikin O., Kochemazov S., Otpuschennikov I., Semenov A., Khamisov O. Problems of search for collisions of cryptographic hash functions of the MD family as variants of Boolean satisfiability problem. *Numerical Methods and Programming*. 2015. vol. 16, no. 1. pp. 61–77. (in Russian)
- 

УДК 004.056.55, 003.26

DOI: 10.14529/cmse170302

## ПОИСК ПРООБРАЗОВ ХЕШ-ФУНКЦИИ MD4 КАК ПРОБЛЕМА ПАРАЛЛЕЛЬНОГО ЛОГИЧЕСКОГО КРИПТОАНАЛИЗА

© 2017 И.А. Грибанова, О.С. Заикин, И.В. Отпущенников,  
А.А. Семёнов

*Институт динамики систем и теории управления  
им. В.М. Матросова СО РАН (664033 Иркутск, ул. Лермонтова, д. 134)  
E-mail: the42dimension@gmail.com, zaikin.icc@gmail.com, otilya@yandex.ru,  
biclop.rambler@yandex.ru*

Поступила в редакцию: 04.05.2017

В статье исследуется задача обращения криптографической хеш-функции MD4, разработанной Р. Ривестом в 1990 году. Через MD4-k обозначается вариант данной функции, в которой параметр k обозначает количество шагов используемых для вычисления хеш-значения (при k=48 имеем полнораундовую версию MD4). В работах Г. Доббертина было показано, что хеш-функция MD4-32 не является односторонней, т.е. для нее может быть решена задача обращения. С этой целью к уравнениям, описывающим конкретные шаги алгоритма вычисления данной функции, были добавлены дополнительные условия на значения некоторых переменных сцепления (chaining variables). Эти дополнительные условия позволили за приемлемое время решить задачу обращения хеш-функции MD4-32 путем решения соответствующей системы уравнений. Основным результатом представляемой статьи является автоматический вывод условий подобных условиям Доббертина (“Dobbertin’s conditions”) при помощи параллельных алгоритмов решения проблемы булевой выполнимости (SAT). Также с использованием данных алгоритмов были решены некоторые задачи обращения функции MD4-k для значений параметра k от 31 до 39 включительно. Стоит отметить, что предложенный метод существенно превосходит по эффективности описанные ранее подходы к решению данной проблемы.

*Ключевые слова: криптоанализ, хеш-функции, задача обращения, MD4, SAT, параллельные вычисления, MPI*

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Gribanova I.A., Zaikin O.S., Otpuschennikov I.V., Semenov A.A. Preimage Attack on MD4 Hash Function as a Problem of Parallel Sat-Based Cryptanalysis // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 16–27. DOI: 10.14529/cmse170302.

### Литература

1. Wang X., Lai X., Feng D., Chen H., Yu X. Cryptanalysis of the Hash Functions MD4 and RIPEMD // Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques. EUROCRYPT’05. Berlin, Heidelberg: Springer-Verlag, 2005. P. 1–18. DOI: 10.1007/11426639\_1.
2. Wang X., Yu H. How to Break MD5 and Other Hash Functions // Proceedings of

- the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques. EUROCRYPT'05. Berlin, Heidelberg: Springer-Verlag, 2005. P. 19–35. DOI: 10.1007/11426639\_2.
3. Dobbertin H. The First Two Rounds of MD4 are Not One-Way // *Fast Software Encryption* / Ed. by Serge Vaudenay. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998. Vol. 1372. P. 284–292. DOI: 10.1007/3-540-69710-1\_19.
  4. Rivest R.L. The MD4 Message Digest Algorithm // *Advances in Cryptology - CRYPTO'90, Proceedings* / Ed. by Alfred Menezes, Scott A. Vanstone. Lecture Notes in Computer Science. Springer, 1990. Vol. 537. P. 303–311. DOI: 10.1007/3-540-38424-3\_22.
  5. Damgård I.B. A Design Principle for Hash Functions // *Proceedings on Advances in Cryptology. CRYPTO '89*. New York, NY, USA: Springer-Verlag New York, Inc., 1989. P. 416–427. DOI: 10.1007/0-387-34805-0\_39.
  6. Merkle R.C. A Certified Digital Signature // *Proceedings on Advances in Cryptology. CRYPTO '89*. New York, NY, USA: Springer-Verlag New York, Inc., 1989. P. 218–238. DOI: 10.1007/0-387-34805-0\_21.
  7. Tseitin G.S. On the Complexity of Derivation in Propositional Calculus // *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983. P. 466–483. DOI: 10.1007/978-3-642-81955-1\_28.
  8. Erkök L., Matthews J. High assurance programming in Cryptol // *Fifth Cyber Security and Information Intelligence Research Workshop, CSIIRW'09*, Knoxville, TN, USA, April 13–15, 2009 / Ed. by Frederick T. Sheldon, Greg Peterson, Axel W. Krings [et al.]. ACM, 2009. P. 60. DOI: 10.1145/1558607.1558676.
  9. Janicic P. URSA: a System for Uniform Reduction to SAT // *Logical Methods in Computer Science*. 2012. Vol. 8, No. 3. P. 1–39. DOI: 10.2168/lmcs-8(3:30)2012.
  10. Soos M., Nohl K., Castelluccia C. Extending SAT Solvers to Cryptographic Problems // *SAT* / Ed. by Oliver Kullmann. Lecture Notes in Computer Science. Springer, 2009. Vol. 5584. P. 244–257. DOI: 10.1007/978-3-642-02777-2\_24.
  11. Otpuschennikov I., Semenov A., Gribanova I., Zaikin O., Kochemazov S. Encoding Cryptographic Functions to SAT Using TRANSALG System // *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August – 2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)* / Ed. by Gal A. Kaminka, Maria Fox, Paolo Bouquet [et al.]. Frontiers in Artificial Intelligence and Applications. IOS Press, 2016. Vol. 285. P. 1594–1595.
  12. Marques-Silva J.P., Sakallah K.A. GRASP: A Search Algorithm for Propositional Satisfiability // *IEEE Trans. Computers*. 1999. Vol. 48, No. 5. P. 506–521. DOI: 10.1109/12.769433.
  13. Marques-Silva J.P., Lynce I., Malik S. Conflict-Driven Clause Learning SAT Solvers // *Handbook of Satisfiability* / Ed. by Armin Biere, Marijn Heule, Hans van Maaren, Toby Walsh. Frontiers in Artificial Intelligence and Applications. IOS Press, 2009. Vol. 185. P. 131–153.
  14. Hyvärinen A.E.J. Grid Based Propositional Satisfiability Solving. Ph.D. thesis, Aalto University, 2011.

15. Mironov I., Zhang L. Applications of SAT Solvers to Cryptanalysis of Hash Functions // SAT / Ed. by Armin Biere, Carla P. Gomes. Lecture Notes in Computer Science. Springer, 2006. Vol. 4121. P. 102–115. DOI: 10.1007/11814948\_13.
16. De D., Kumarasubramanian A., Venkatesan R. Inversion Attacks on Secure Hash Functions Using SAT Solvers // Theory and Applications of Satisfiability Testing - SAT 2007, Proceedings / Ed. by João Marques-Silva, Kareem A. Sakallah. Lecture Notes in Computer Science. Springer, 2007. Vol. 4501. P. 377–382. DOI: 10.1007/978-3-540-72788-0\_36.
17. Eén N., Sörensson N. Temporal Induction by Incremental SAT Solving // Electr. Notes Theor. Comput. Sci. 2003. Vol. 89, No. 4. P. 543–560. DOI: 10.1016/s1571-0661(05)82542-3.
18. Semenov A., Zaikin O. Algorithm for Finding Partitionings of Hard Variants of Boolean Satisfiability Problem with Application to Inversion of Some Cryptographic Functions // SpringerPlus. 2016. Vol. 5, No. 1. P. 1–16. DOI:10.1186/s40064-016-2187-4.
19. Bogachkova (Gribanova) I., Zaikin O., Kochemazov S., Otpuschennikov I., Semenov A., Khamisov O. Problems of Search for Collisions of Cryptographic Hash Functions of the MD Family as Variants of Boolean Satisfiability Problem // Numerical Methods and Programming. 2015. Vol. 16, No. 1. P. 61–77. (in Russian)

Грибанова Ирина Александровна, аспирант, лаборатория 3.2, Институт динамики систем и теории управления имени В.М. Матросова СО РАН (Иркутск, Российская Федерация)

Заикин Олег Сергеевич, к.т.н., с.н.с., лаборатория 3.2, Институт динамики систем и теории управления имени В.М. Матросова СО РАН (Иркутск, Российская Федерация)

Отпущенников Илья Владимирович, к.т.н., н.с., лаборатория 3.2, Институт динамики систем и теории управления имени В.М. Матросова СО РАН (Иркутск, Российская Федерация)

Семёнов Александр Анатольевич, доцент, к.т.н., зав. лабораторией 3.2, Институт динамики систем и теории управления имени В.М. Матросова СО РАН (Иркутск, Российская Федерация)

## ОБЗОР МЕТОДОВ ОБУЧЕНИЯ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

© 2017 А.В. Созыкин

*Институт математики и механики им. Н.Н.Красовского УрО РАН*

*(620990 Екатеринбург, ул. Софьи Ковалевской, д. 16),*

*Уральский федеральный университет*

*(620002 Екатеринбург, ул. Мира, д. 19)*

*E-mail: avs@imm.uran.ru*

Поступила в редакцию: 12.04.2017

Глубокие нейронные сети в настоящее время становятся одним из самых популярных подходов к созданию систем искусственного интеллекта, таких как распознавание речи, обработка естественного языка, компьютерное зрение и т.п. В статье представлен обзор истории развития и современного состояния методов обучения глубоких нейронных сетей. Рассматривается модель искусственной нейронной сети, алгоритмы обучения нейронных сетей, в том числе алгоритм обратного распространения ошибки, применяемый для обучения глубоких нейронных сетей. Описывается развитие архитектур нейронных сетей: неокогнитрон, автокодировщики, сверточные нейронные сети, ограниченная машина Больцмана, глубокие сети доверия, сети долго-краткосрочной памяти, управляемые рекуррентные нейронные сети и сети остаточного обучения. Глубокие нейронные сети с большим количеством скрытых слоев трудно обучать из-за проблемы исчезающего градиента. В статье рассматриваются методы решения этой проблемы, которые позволяют успешно обучать глубокие нейронные сети с более чем ста слоями. Приводится обзор популярных библиотек глубокого обучения нейронных сетей, которые сделали возможным широкое практическое применение данной технологии. В настоящее время для задач компьютерного зрения используются сверточные нейронные сети, а для обработки последовательностей, в том числе естественного языка, — рекуррентные нейронные сети, прежде всего сети долго-краткосрочной памяти и управляемые рекуррентные нейронные сети.

*Ключевые слова: глубокое обучение, нейронные сети, машинное обучение.*

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Созыкин А.В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 28–59. DOI: 10.14529/cmse170303.

### Введение

Глубокие нейронные сети в настоящее время становятся одним из самых популярных методов машинного обучения. Они показывают лучшие результаты по сравнению с альтернативными методами в таких областях, как распознавание речи, обработка естественного языка, компьютерное зрение [1], медицинская информатика [2] и др. Одна из причин успешного применения глубоких нейронных сетей заключается в том, что сеть автоматически выделяет из данных важные признаки, необходимые для решения задачи. В альтернативных алгоритмах машинного обучения признаки должны выделяться людьми, существует специализированное направление исследований — *инженерия признаков* (feature engineering). Однако при обработке больших объемов данных нейронная сеть справляется с выделением признаков гораздо лучше, чем человек.

В статье представлен исторический обзор развития архитектур глубоких нейронных сетей и подходов к их обучению. Задача составления такого обзора существенно

затруднена тем, что вариантов глубоких нейронных сетей было предложено очень много и терминология менялась со временем [3]. Модель искусственных нейронных сетей была предложена в 1943 году [4], а сам термин *глубокое обучение* (deep learning) стал широко использоваться только начиная с 2006 года [5, 6]. До этого применялись термины *загрузка глубоких сетей* (loading deep networks) [7, 8] и *обучение глубокой памяти* (learning deep memories) [9].

Рост популярности глубоких нейронных сетей, происходящий в последние несколько лет, можно объяснить тремя факторами. Во-первых, произошло существенное увеличение производительности компьютеров, в том числе ускорителей вычислений GPU (Graphics Processing Unit), что позволило обучать глубокие нейронные сети значительно быстрее и с более высокой точностью [10]. Ранее имеющихся вычислительных мощностей не хватало для обучения сколько-нибудь сложной сети, пригодной для решения практических задач. Во-вторых, был накоплен большой объем данных, который необходим для обучения глубоких нейронных сетей. В-третьих, разработаны методы обучения нейронных сетей, позволяющие быстро и качественно обучать сети, состоящие из ста и более слоев [11], что раньше было невозможно из-за проблемы исчезающего градиента и переобучения. Сочетание трех факторов привело к существенному прогрессу в обучении глубоких нейронных сетей и их практическом использовании, что позволило глубоким нейронным сетям занять лидирующую позицию среди методов машинного обучения.

Статья организована следующим образом. В разделе 1 рассмотрена модель искусственных нейронных сетей. Раздел 2 посвящен методам обучения нейронных сетей. В разделе 3 приведен исторический обзор развития архитектур глубоких нейронных сетей. В разделе 4 рассматриваются популярные программные системы для обучения глубоких нейронных сетей. В заключении подводятся итоги и описываются перспективы развития архитектур и методов обучения глубоких нейронных сетей.

## 1. Искусственные нейронные сети

Модель искусственного нейрона была предложена Уорреном МакКаллоком (Warren McCulloch) и Уолтером Питтсом (Walter Pitts) в 1943 году в работе [4]. В качестве основы для своей модели авторы использовали биологический нейрон. Искусственный нейрон МакКаллока—Питтса имеет  $N$  входных бинарных величин  $x_1, \dots, x_n$ , которые трактуются как импульсы, поступающие на *вход нейрону* (рис. 1). В нейроне импульсы складываются с *весами*  $w_1, \dots, w_n$ .

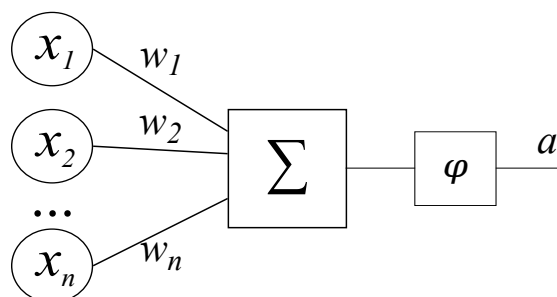


Рис. 1. Модель искусственного нейрона МакКаллока—Питтса

Выходной сигнал нейрона определяется по формуле:

$$a = \varphi\left(\sum_{i=1}^N w_i x_i\right), \quad (1)$$

где нелинейная функция  $\varphi$  (*функция активации*) преобразует суммарный импульс в выходное значение нейрона. В модели МакКаллока—Питтса для этой цели использовалась функция Хевисайда. В дальнейшем было предложено использовать другие типы функций активации: логистическую сигмоидальную ( $f(x) = \frac{1}{1+e^{-x}}$ ) [12], гиперболический тангенс ( $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$ ) [13] и радиально-базисную функцию [14]. Такие функции активации обеспечивали более плавное изменение выходного сигнала нейрона.

МакКаллок и Питтс предложили также метод объединения отдельных нейронов в *искусственные нейронные сети*. Для этого выходные сигналы нейрона передаются на вход следующему нейрону (рис. 2). Нейронная сеть состоит из нескольких *слоев*, на каждом из которых может находиться несколько нейронов. Слой, который принимает сигналы из внешнего мира, называется *входным*. Слой, который выдает сигналы во внешний мир, — *выходным*. Остальные слои называются *скрытыми*.

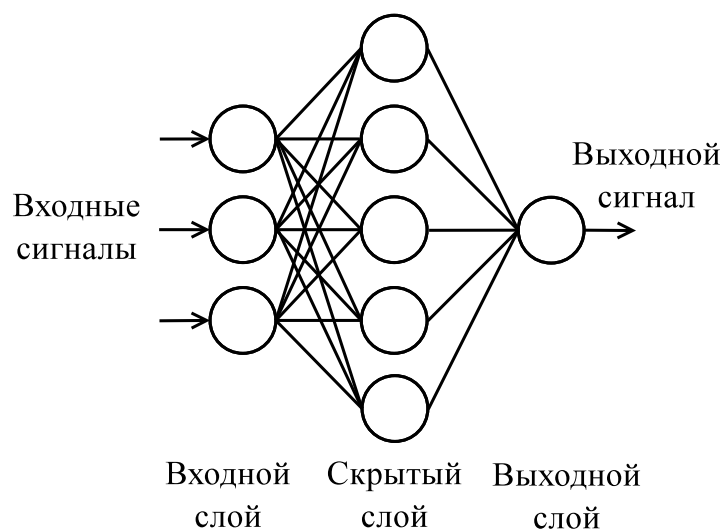


Рис. 2. Искусственная нейронная сеть

Искусственные нейронные сети делятся на *сети прямого распространения сигнала* (feedforward networks), в которых нет циклов, и *рекуррентные сети* (recurrent networks), в которых циклы разрешены.

Однозначного определения, что такое *глубокая нейронная сеть*, не существует. В данной статье под глубокой нейронной сетью будет пониматься такая нейронная сеть, которая содержит более одного скрытого слоя.

Искусственные нейронные сети, заданные таким образом, способны приблизить любую непрерывную функцию с любой требуемой точностью [15–18]. Однако в настоящее время не существует конструктивного подхода, который бы позволял гарантированно создавать нейронные сети с заранее заданными свойствами. Это является существенным недостатком, ограничивающим применение нейронных сетей.

## 2. Обучение нейронных сетей

*Обучение* нейронной сети — это процесс определения весов соединений между нейронами таким образом, чтобы сеть приближала необходимую функцию с заданной точностью. Существует три подхода к обучению нейронных сетей [3]: *обучение с учителем* (supervised learning), *обучение без учителя* (unsupervised learning) и *обучение с подкреплением* (reinforcement learning). При обучении с учителем на вход сети подаются наборы входных сигналов (*объектов*), для которых заранее известен правильный ответ (*обучающее множество*). Веса меняются по определенным правилам в зависимости от того, правильный ли выходной сигнал выдала сеть. При обучении без учителя на вход сети подаются объекты, для которых правильный выходной сигнал заранее не известен. Обучение с подкреплением предполагает наличие внешней среды, с которой взаимодействует сеть. Обучение происходит на основании сигналов, полученных от этой среды.

Нейронные сети МакКаллока—Питтса не обучались. Веса для всех входов нейронов должны были быть заданы заранее.

Впервые идею обучения нейронных сетей предложил Дональд Хэбб (Donald Hebb) в 1949 году [19]. Согласно Хэббу, связи нейронов, которые активируются вместе, должны усиливаться, а связи нейронов, которые срабатывают отдельно друг от друга, должны ослабевать. Хэбб предложил правила изменения веса входных сигналов нейронов в соответствии с тем, правильный ответ выдавала сеть, или нет [19] (обучение с учителем). А.В. Новиков доказал сходимость предложенного метода обучения нейрона на основе правил Хэбба [20], при условии, что выборка объектов линейно разделима. Впоследствии было предложено несколько аналогичных правил как для обучением с учителем [21–23], так и без учителя [24–28].

В 1970 году А.Г. Ивахненко разработал *метод группового учета аргументов* (group method of data handling) [29, 30], позволяющий не только вычислять веса связей между нейронами, но и определять количество слоев в сети и нейронов в них в зависимости от потребностей прикладной задачи. Используя подход обучения с учителем, уровни сети инкрементально строятся и обучаются на основе обучающего множества с использованием регрессионного анализа. Затем происходит этап упрощения сети с применением отдельного множества объектов с известными правильными ответами, которое не использовалось при обучении (*проверочное множество*, validation set). Для исключения ненужных нейронов из сети используется регуляризация. В работе [30] описано применение метода группового учета аргументов для обучения глубокой нейронной сети, состоящей из восьми слоев. Метод широко использовался на практике [31–33].

В настоящее время для обучения нейронных сетей, в том числе глубоких, используется алгоритм *обратного распространения ошибки* (error backpropagation algorithm), основанный на методе градиентного спуска. Алгоритм был предложен в 1970 году в магистерской диссертации [34, 35] без связи с нейронными сетями. Первое применение этого алгоритма для обучения нейронных сетей описано в работе [36], вышедшей в 1981 году. После этого появилось еще несколько работ на эту тему [37–39]. Алгоритм обратного распространения ошибки использует обучение с учителем, для него требуется обучающее множество с заранее известными правильными ответами. Вводится мера ошибки, которая определяет, насколько сильно выходные значения сети отличаются от правильных ответов. Затем мера ошибки минимизируется с помощью метода градиентного спуска путем

изменения значений весов в сети. Для того чтобы оценить, насколько сильно каждый вес влияет на выходное значение, рассчитываются частные производные ошибки по весам. Затем производится изменение весов на небольшие значение с учетом градиента. Так повторяется до тех пор, пока ошибка на выходе не сократится до допустимых значений. Начальные значения весов нейронов в сети задаются случайным образом.

В глубокой нейронной сети с несколькими скрытыми слоями производится расчет ошибки, которая передается от одного слоя к другому. На первом этапе рассчитывается значение ошибки на выходе нейронной сети, для которого мы знаем правильные ответы. Затем рассчитывается ошибка на входе в выходной слой сети, которая будет использоваться как ошибка на выходе скрытого слоя. Таким способом расчет продолжается до того момента, когда будет известна ошибка на входном слое. Именно поэтому алгоритм имеет название обратное распространение ошибки.

Возможно несколько вариантов реализации обучения нейронных сетей с помощью алгоритма обратного распространения ошибки. При полном обучении градиент рассчитывается для всех объектов обучающего множества. Однако такой подход часто не является эффективным в случае, когда обучающее множество большое и для обработки всех его элементов требуется значительное время. Альтернативный вариант — использование метода стохастического градиентного спуска, при котором веса изменяются при обработке одного элемента обучающего множества (онлайн-обучение) или нескольких элементов (обучение на пакетах или мини-выборках). На практике для обучения нейронных сетей чаще всего используется именно метод стохастического градиентного спуска или его модификации [40–42].

### 3. Архитектура глубоких нейронных сетей

Нейронная сеть, показанная на рис. 2, называется *полносвязной*. В такой сети каждый нейрон следующего слоя связан со всеми нейронами предыдущего слоя. Однако это не единственный вариант соединения нейронов в сеть. В данном разделе рассматривается развитие архитектур нейронных сетей.

В 1980 году Кунихико Фукушима (Kunihiko Fukushima) предложил архитектуру нейронной сети, которая называется *неокогнитрон* [43]. Архитектура использовала аналогию со сложными и простыми клетками в зрительной коре кошки [44]. Простые клетки срабатывают в ответ на простые визуальные сигналы, такие как ориентация границ. Сложные клетки менее зависимы от пространственного расположения сигналов и ориентируются на более общие признаки. В неокогнитроне простым клеткам соответствуют *сверточные слои* (convolutional layers), а сложным — *слои подвыборки* (subsampling layers). В сверточных слоях окно *сверточного узла* (convolutional unit) с заданным набором весов (*ядро свертки*) перемещается по двумерному массиву входных данных, например, по пикселям изображения (рис. 3). Значения совпадающих элементов в данных и ядре свертки перемножаются, полученные результаты складываются и поступают в нейрон следующего слоя. Все сверточные узлы используют одинаковые ядра свертки, поэтому для описания сверточной сети требуется относительно немного параметров. Как правило, в сверточных слоях используется не одно, а несколько ядер свертки.

Выходы сверточных слоев в неокогнитроне подключаются ко входам слоев подвыборки. Причем к одному нейрону в слое подвыборки подключаются несколько нейронов сверточного слоя, как правило из квадратной области размером  $2 \times 2$  или больше. Нейроны



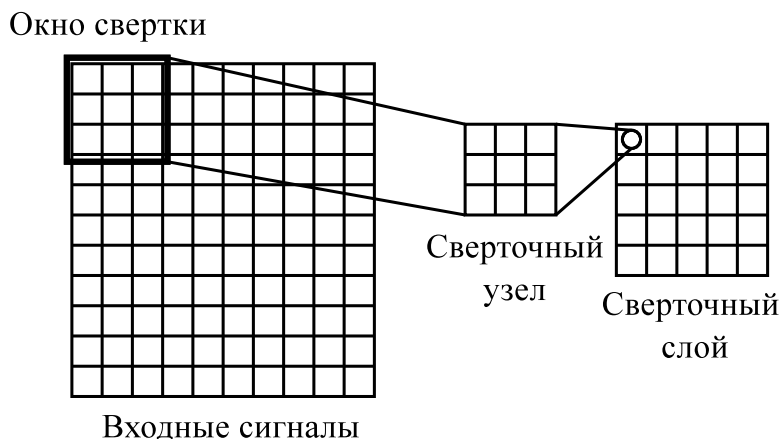


Рис. 3. Схема сверточного слоя нейронной сети неокогнитрон

в слое подвыборки срабатывают в случае активности хотя бы один из входных сигналов. На этом слое важно наличие самого сигнала, а не его конкретные координаты, поэтому слои подвыборки менее чувствительны к незначительным сдвигам и изменениям масштаба изображения. Обучение сверточных слоев в неокогнитроне производится с помощью локальных алгоритмов обучения без учителя, либо веса задаются заранее [45, 46]. Для слоев подвыборки используется *пространственное усреднение* (spatial averaging) [43, 47]. Таким образом, несмотря на то, что неокогнитрон является глубокой нейронной сетью, глубокое обучение в нем не используется.

В 1987 году Дана Баллард (Dana Ballard) предложил подход к обучению нейронных сетей без учителя на основе *автокодировщика* (autoencoder) [48]. Простой автокодировщик содержит всего один скрытый слой (рис. 4) с кодом  $h$ , который служит для представления входного сигнала  $x$ . Автокодировщик содержит функцию кодирования  $f$ , которая используется для преобразования входного сигнала в код  $h = f(x)$  и функцию декодирования  $g$ , которая по коду восстанавливает значения входных сигналов  $r = g(h)$ .

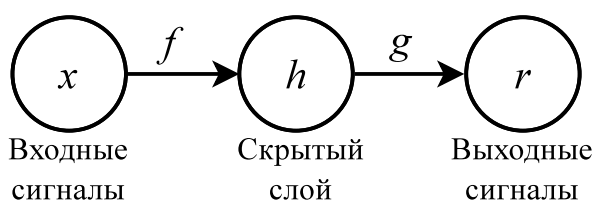


Рис. 4. Схема простого автокодировщика

Автокодировщики применяются для уменьшения размерности обрабатываемых данных. Для этого используются линейные методы, такие как метод главных компонент. За счет снижения размерности автокодировщик выделяет наиболее значимые характеристики данных.

Для обучения автокодировщиков используется метод *рециркуляции* [49]. Автокодировщик обучается таким образом, чтобы на его выходе были те же самые сигналы, как и на входе. После обучения скрытый слой простого автокодировщика вставляется в автокодировщик более высокого уровня, который содержит больше скрытых слоев. Таким образом строится иерархия автокодировщиков в виде стека (рис. 5). При этом размерность данных уменьшается на каждом уровне иерархии. Подобная иерархия может

быть использована не только для автокодировщиков, но и для других методов обучения без учителя [50, 51].

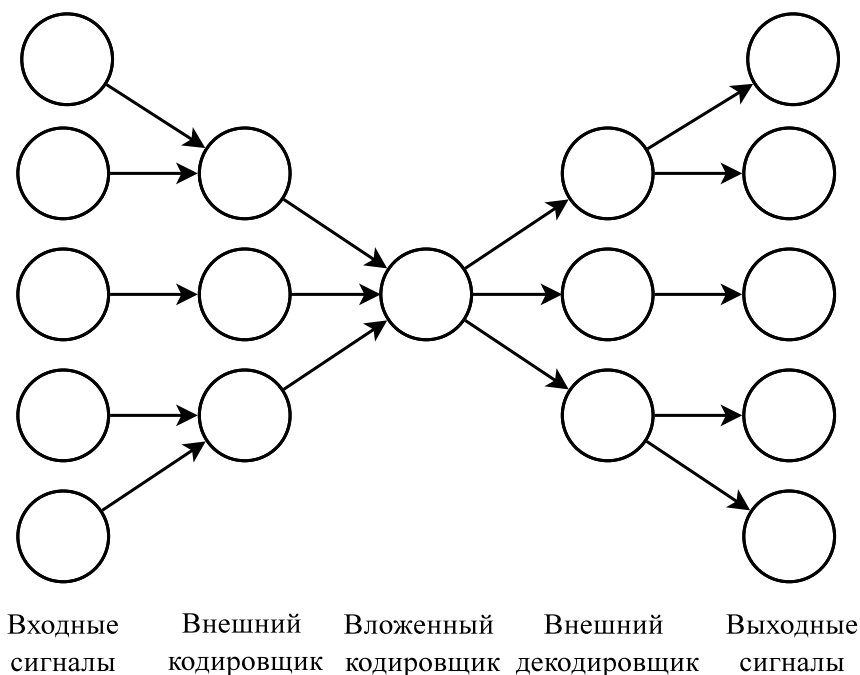


Рис. 5. Иерархия автокодировщиков

Ян Лекун (Yann LeCun) в 1989 году применил алгоритм обратного распространения ошибки для обучения сети с архитектурой, очень похожей на неокогнитрон [52]. Сеть содержала сверточные слои с одинаковыми весами (сверточными ядрами) и слои подвыборки. В этой же работе был представлен набор данных MNIST, содержащий рукописные цифры, распознавание которых со временем стало очень популярным тестом в машинном обучении. Глубокие сверточные сети, обученные алгоритмом обратного распространения ошибки, показали хорошие результаты на распознавании рукописных цифр [53] и отпечатков пальцев [54].

В конце 80-х годов XX века стало понятно, что одного алгоритма обратного распространения ошибки недостаточно для эффективного глубокого обучения. Несмотря на некоторые успешные примеры [53, 54], использование более одного скрытого слоя редко давало преимущества на практике [55–57]. Причина этого была сформулирована в 1991 году в работе [58, 59] — *проблема исчезающего градиента* (vanishing gradient problem). При использовании традиционных функций активации (см. раздел 1), сигналы об обратном распространяемых ошибках быстро становятся очень маленькими (или наоборот, чрезмерно большими). В практических задачах они уменьшаются экспоненциально с количеством слоев в сети. Эта проблема также известна как *проблема длительной задержки* (long time lag problem) [60, 61].

Один из подходов к решению проблемы исчезающего градиента заключается в полном отказе от использования градиента для обучения. Для некоторых задач хороших результатов можно добиться назначением весов случайным образом [62]. Подход к обучению *Еволино* (Evolino) [63] использует линейные методы для определения оптимальных весов для выходного слоя и *эволюцию* для определения весов скрытых слоев. Также возможно применение методов *универсального поиска* [64, 65].

Альтернативный подход к решению проблемы исчезающего градиента — использование безгессианной оптимизации (Hessian-free optimization) [66–70].

*Очень глубокий обучатель*, предложенный в работе 1992 года [71], обеспечивает решение проблемы исчезающего градиента и обучение сети глубиной до сотен слоев за счет использования предварительного обучения без учителя иерархии рекуррентных нейронных сетей. Каждая рекуррентная сеть обучается отдельно для того, чтобы предсказать следующее значение, которое поступит ей на вход [72, 73]. После обучения только ошибочно предсказанные значения передаются на более высокий уровень сети. Эта сеть работает уже на более медленной временной шкале, за счет чего информация сжимается и каждой последовательности сигналов соответствует набор все менее и менее избыточного кодирования на все более глубоких уровнях сети. Другое название такой архитектуры нейронной сети — *компрессор истории* (History Compressor), она может сжимать данные как в пространстве, так и во времени. Существует также и непрерывный вариант компрессора истории [74].

Проблему исчезающего градиента позволяет решить другая архитектура рекуррентной нейронной сети — *сеть долго-краткосрочной памяти* (Long Short-Term Memory) [75–77]. Такие сети содержат узлы специального типа, которые позволяют запоминать значения на длительный срок. Блок сети долго-краткосрочной памяти содержит специальный нейрон, используемый в качестве ячейки памяти (рис. 6). Выход нейрона соединен с его собственным входом с единичным весом. За счет этого значение в нейроне на каждом этапе перезаписывается и таким образом сохраняется. Управление нейроном выполняется с помощью трех вентилях: входного, выходного и вентиля забвения. При открытом входном вентиле значение на входе записывается в ячейку памяти. Если входной вентиль закрыт, то входные сигналы никак не влияют на содержимое ячейки. Открытый выходной вентиль позволяет прочитать значение из ячейки. Когда значение больше не нужно, его можно стереть с помощью вентиля забвения. Вентили подключаются к другим узлам нейронной сети, которые в процессе обучения определяют, когда необходимо открыть или закрыть тот или иной вентиль.

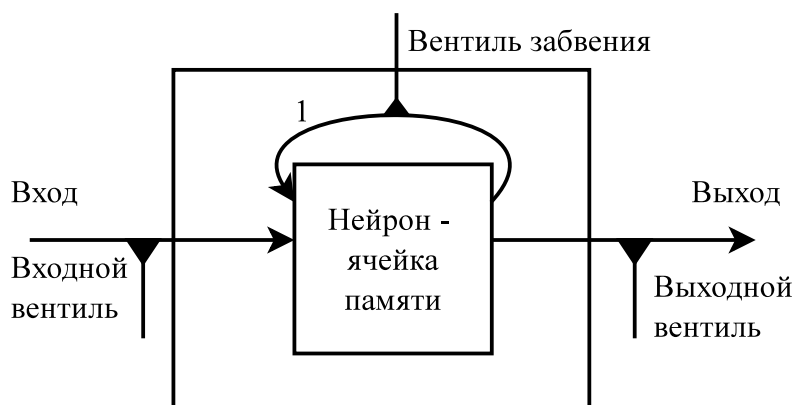


Рис. 6. Схема блока сети долго-краткосрочной памяти

Благодаря таким ячейкам сети долго-краткосрочной памяти могут определить важность событий, произошедших тысячи дискретных временных шагов назад, и запомнить эти события. Рекуррентные сети, которые использовались до этого, могли помнить о событии не дольше десяти временных шагов. Проблема исчезающего градиента в сетях долго-краткосрочной памяти решается за счет использования функции тождества

в качестве активационной и обратной связи с собственным входом с весом равным единице (рис. 6). Так как производная функции тождества равна единице, то ошибка при передаче через такие узлы не может исчезнуть.

В 1992 году появилась архитектура нейронных сетей *кресцентрон* (crescetrone) [78], основой для которой послужил неокогнитрон. В отличие от неокогнитрона, кресцентрон изменяет свою топологию во время обучения, по аналогии с сетями, использующими метод группового учета аргументов [29]. Важная идея, предложенная в кресцентроне — использование слоев *выбора максимального элемента* (max-pooling) вместо слоев подвыборки с усреднением. Более поздние и усовершенствованные версии Кресцентрона содержали также слои *размывания* (blurring) для уменьшения зависимости от положения объектов [79]. Слои максимального выбора сейчас широко применяются в сверточных нейронных сетях. Однако для обучения современных сверточных сетей используется алгоритм обратного распространения ошибки [80], что является более эффективным [81].

В 2006 году Джеффри Хинтон (Geoffrey Hinton) и Руслан Салахутдинов (Ruslan Salakhutdinov) предложили новую архитектуру нейронных сетей — *глубокие сети доверия* (Deep Belief Networks) [5]. В этой архитектуре для решения проблемы исчезающего градиента использовалась комбинация обучения с учителем и без учителя. Глубокая сеть доверия представляет собой стек *ограниченных машин Больцмана* [82, 83]. Каждая такая машина содержит только два слоя нейронов: входной и скрытый (рис. 7). Соединения есть только между нейронами разных слоев, в отличие от машины Больцмана высокого порядка [84], которая может содержать другие типы связей.

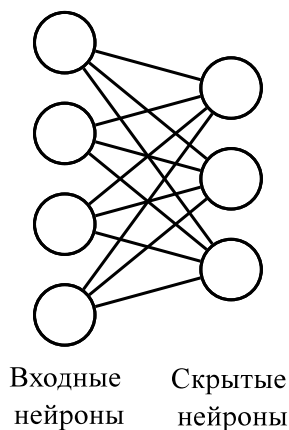


Рис. 7. Ограниченная машина Больцмана

Каждая ограниченная машина Больцмана получает сигналы о представлении шаблонов с предыдущего уровня и пытается закодировать их с использованием обучения без учителя [6] (рис. 7). После этого производится тонкая настройка всей сети с помощью обучения с учителем с использованием алгоритма обратного распространения ошибки. Обученная таким образом глубокая сеть доверия показала хорошие результаты на тесте распознавания рукописных цифр MNIST [5], распознавании фонем [85], поиске документов [86] и других задачах. Похожим альтернативным подходом, основанным на предварительном обучении без учителя и последующей точной настройке путем обучения с учителем, является использование стека автокодировщиков [48, 87–90].

В 2006–2007 годах также произошло развитие глубокого обучения сверточных сетей на основе обучения с учителем без предварительного обучения без учителя. В работе [80] впервые описано применение алгоритма обратного распространения ошибки для обучения

глубокой нейронной сети с архитектурой, подобной неокогнитрону и кресцептрону, состоящей из чередующихся слоев свертки и максимального выбора. Такая архитектура нейронных сетей активно используется по настоящее время [91–95].

Существенный вклад в развитие сверточных нейронных сетей внесло предложение использовать *полулинейную функцию активации* (rectified linear unit) [96, 97], которая задается следующим образом:  $ReLU(x) = \max(0, x)$ . Такая функция активации позволяет избавиться от проблемы исчезающего градиента, т.к. при положительном значении сигнала не происходит его изменения, в отличие от сигмоидальных функций активации. Кроме того, полулинейная функция активации позволяет сократить время обучения нейронной сети [98]. Нейроны, выходной сигнал которых отрицательный, не участвуют в расчетах, а для остальных требуется выполнять только линейные вычисления.

В 2010 году Ксавье Глоро (Xavier Glorot) и Йошуа Бенджио (Yoshua Bengio) в работе [99] провели исследование влияния методов начальной инициализации весов и функций активации на распространение сигнала в сети как в прямом, так и в обратном направлении. Согласно исследованию, использование логистической сигмоидальной функции совместно с начальной инициализацией весов плохо подходит для создания глубоких нейронных сетей, т.к. приводит к быстрому насыщению. Функция активации гиперболический тангенс такой проблемой не обладает из-за симметричности (среднее значение функции 0, область значения  $-(1, 1)$ ). Глоро и Бенджио предложили новый метод инициализации весов нейронной сети, который они назвали *нормализованная инициализация*. Начальные значения весов сети  $W$  определяются по следующей формуле:

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right], \quad (2)$$

где  $U$  — равномерное распределение на отрезке,  $n_j$  — количество нейронов на текущем слое сети,  $n_{j+1}$  — количество нейронов на следующем слое сети. Использование нормализованной инициализации приводит к снижению насыщения нейронов и сигнал об ошибке распространяется значительно лучше.

Метод нормализованной инициализации весов в 2015 году был адаптирован для полулинейной функции активации. В работе [100] предлагается определять начальные веса  $W$  следующим образом:

$$W \sim U\left[-\frac{2}{n_j}, \frac{2}{n_j}\right], \quad (3)$$

где  $U$  — равномерное распределение на отрезке,  $n_j$  — количество нейронов на текущем слое сети. Метод нормализованной инициализации весов позволил достигать качественного обучения глубоких нейронных сетей без необходимости использовать предварительное обучение без учителя.

Сергей Йоффе (Sergey Ioffe) и Кристиан Жегеды (Christian Szegedy) в 2015 году [101] предложили использовать в нейронных сетях специальные слои пакетной нормализации (batch normalization), которые позволяют повысить качество обучения глубокой нейронной сети. В работе [13] установлено, что алгоритм обратного распространения ошибки сходится быстрее, если входные данные нормализованы (имеют нулевое матожидание и единичную дисперсию). Однако при распространении сигнала по нейронной сети его матожидание и дисперсия меняются, причем иногда значительно, что негативно сказывается на процессе обучения. Сергей Йоффе и Кристиан Жегеды предложили выполнять нормализацию не только на входе в нейронную сеть, но и перед каждым слоем сети. Нормализация

выполняется отдельно для каждого *пакета данных* (mini-batch) метода стохастического градиентного спуска (или его модификаций). Пакет  $B$  содержит  $m$  элементов входных данных  $x_i$ :  $B = \{x_1, \dots, x_m\}$ . Нормализованные значения  $\hat{x}_i$  определяются по следующей формуле:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad (4)$$

где  $\mu_B$  — среднее значение данных в пакете,  $\sigma_B^2$  — дисперсия,  $\epsilon$  — константа, введенная для стабильности метода. Затем, чтобы сохранить выразительность данных, выполняется сдвиг и масштабирование:

$$y_i = \gamma \hat{x}_i + \beta, \quad (5)$$

где  $y_i$  — результирующее значение,  $\gamma$  и  $\beta$  — параметры, которые определяются в процессе обучения.

Пакетная нормализация реализуется в виде слоев пакетной нормализации, которые могут быть вставлены в необходимое место в нейронной сети, в том числе несколько раз. Дополнительным преимуществом использования пакетной нормализации является сокращение времени обучения и снижение переобучения.

Методы нормализованной инициализации весов и слои пакетной нормализации помогают на практике справиться с проблемой исчезающего градиента и обучать глубокие нейронные сети, состоящие из нескольких десятков слоев. Это позволило некоторым авторам утверждать, что проблема исчезающего градиента в настоящее время решена [11].

Компания Google в 2014 году предложила новую архитектуру сверточных нейронных сетей *Inception* [102]. В этой архитектуре сеть строится из набора блоков, содержащих комбинацию операций свертки и подвыборки разной размерности. Такой подход позволяет избежать переобучения, а также снизить количество параметров сети, которые необходимо обучать, что снижает время обучения сети. Первая версия блока сети архитектуры Inception показана на рис. 8. Сеть строится из нескольких повторяющихся блоков Inception, которые следуют друг за другом.



Рис. 8. Блок сети Google Inception [102]

На основе предложенной архитектуры была создана и успешно обучена сеть *GoogLeNet* [102], состоящая из 22 слоев. Сеть применяется для задач компьютерного зрения. Впоследствии Google предложила еще несколько вариантов блоков архитектуры Inception и сетей на ее основе [103, 104].

В 2014 году также была предложена новая архитектура рекуррентных нейронных сетей — управляемые рекуррентные нейронные сети (gated recurrent neural networks) [105, 106]. Такие сети похожи на сети долго-краткосрочной памяти, но в них не используются ячейки памяти. Схема блока управляемой рекуррентной сети показана на рис. 9.

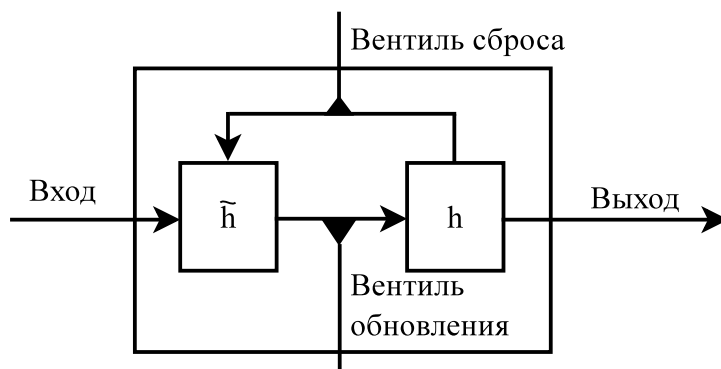


Рис. 9. Блок управляемой рекуррентной нейронной сети [105]:  $h$  — значение в блоке,  $\tilde{h}$  — новое значение

Блок управляемой нейронной сети, в отличие от сетей долго-краткосрочной памяти, не содержит выходного вентиля, поэтому значение в блоке всегда выдается наружу. Используется два типа вентиля: обновления и сброса. Вентиль обновления определяет, будет ли применяться новое значение  $\tilde{h}$ , а вентиль сброса задает, учитывается ли текущее значение  $h$  при расчете нового значения. Управляемые нейронные сети хорошо показали себя в задачах моделирования сигналов речи и полифонической музыки, а также для автоматического перевода [106, 107].

Увеличение количества слоев в нейронных сетях, даже с использованием полулинейных функций активации, нормализации начальных значений весов и слоев нормализации, не всегда приводит к увеличению качества обучения. При этом причина не в проблеме исчезающего градиента и не в переобучении, а в увеличении ошибки обучения сети при росте количества слоев [11, 108]. Для решения этой проблемы компания Microsoft в 2015 году предложила новую архитектуру и подход к обучению нейронных сетей — *остаточное обучение* (residual learning). Архитектура использует тот факт, что нейронную сеть всегда можно сделать более глубокой без снижения качества работы путем добавления нескольких слоев, которые не меняют сигнал. Сеть остаточного обучения, как и сеть Google Inception, строится из повторяющихся блоков. Схема блока показана на рис. 10. Блок включает два обычных слоя нейронной сети, веса в которых обучаются с помощью алгоритма обратного распространения ошибки, а также параллельный им тождественный слой, не изменяющий входных сигналов. Тождественный слой используется, если не получается обучить основные слои. Таким образом, сеть в процессе обучения сама определяет, сколько слоев нужно для решения задачи.

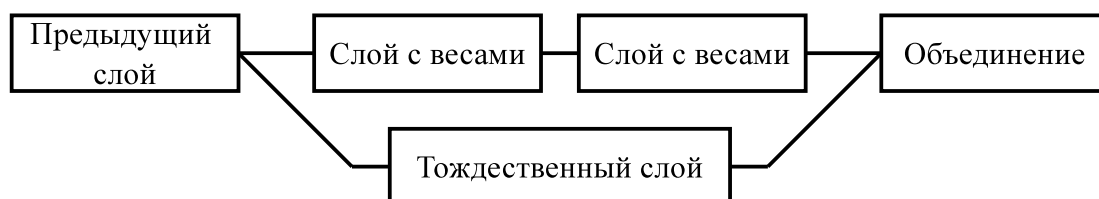


Рис. 10. Блок сети остаточного обучения [11]

Компания Microsoft удалось собрать из блоков сети остаточного обучения глубиной 34, 50, 101 и даже 152 слоя, успешно обучить их и применить для решения различных задач компьютерного зрения [11]. Компания Google также стала использовать подход остаточного обучения в сетях архитектуры Inception [104].

#### 4. Программные системы обучения глубоких нейронных сетей

В настоящее время создано большое количество программных систем для обучения глубоких нейронных сетей [109–111]. Среди наиболее популярных из них можно отметить Caffe, Theano, TensorFlow, Torch и CNTK. Их основные характеристики приведены в табл.

Библиотека Caffe [109] — одна из самых первых популярных систем глубокого обучения. Ее разработали в центре компьютерного зрения и обучения в Беркли (Berkeley Vision and Learning Center), исходные коды стали открытыми в 2014 году. Caffe включает самое большое количество готовых к использованию предварительно обученных моделей. Система Theano [112] создана в Университете Монреаля, Канада. Theano разработана на Python, но обеспечивает высокую производительность за счет того, что программа на Python автоматически преобразуется в программу на C++, которая компилируется и затем выполняется. TensorFlow [113] создана компанией Google в 2015 году и включает системы эффективной работы с тензорами и потоковой обработки данных на графе. Библиотека Torch [114] разработана на языке Lua и предоставляет удобный высокоуровневый интерфейс для создания программ машинного обучения, аналогичный MATLAB. Высокая производительность обеспечивается, так же как и в Theano, за счет интеграции с языком C. Авторы Torch предпочли использовать Lua вместо Python из-за простоты интеграции C и Lua. Компания Microsoft разработала систему CNTK (Cognitive Toolkit) [115] и открыла ее исходные коды в 2016 году.

Все перечисленные системы глубокого обучения нейронных сетей могут использовать для ускорения обучения как многоядерные процессоры, так и ускорители вычислений GPU (включая оптимизированную библиотеку cuDNN). Причем существенным преимуществом является то, что нет необходимости переделывать программу, распараллеливание на CPU и GPU выполняется автоматически. Системы Caffe и Theano дополнительно поддерживают ускорители Intel Xeon Phi, которые также помогают существенно сократить время обучения глубоких нейронных сетей [116]. Почти все системы, кроме Theano, можно использовать для распределенного обучения нейронных сетей на вычислительных кластерах.

В дополнение к описанным выше системам можно отметить также библиотеку Keras [117], которая предоставляет удобный и простой в использовании программный интерфейс для обучения глубоких нейронных сетей. Keras не является самостоятельной системой, а работает поверх Theano, TensorFlow или CNTK. В 2016 году Keras включили в состав TensorFlow.

Заслуживают внимания также и новые библиотеки глубокого обучения, созданные недавно, но набирающие популярность. Системы PaddlePaddle [118] (создана компанией Baidu) и MXNet [119] специально разработаны для обучения глубоких нейронных сетей на распределенных кластерах. Библиотека Neon [120] разрабатывалась компанией Nervana. После покупки Nervana компанией Intel, Neon стала одной из самых быстро развивающихся библиотек с поддержкой ускорителей GPU и Intel Xeon Phi, а также большим количеством



Таблица

Программные системы обучения глубоких нейронных сетей

Свойство	Caffe	Theano	TensorFlow	Torch	CNTK
Базовый язык	C++	Python	C++	Lua	C++
API	C++ Python	Python	C++ Python	Lua Python	C++, C# Python
Многоядерные CPU	+	+	+	+	+
GPU	+	+	+	+	+
Xeon Phi	+	+	–	–	–
Распределенное обучение	+	–	+	+	+
Разработчик	Центр компьютерного зрения и обучения Беркли	Университет Монреаля	Google	Ронан Коллаберт	Microsoft
Открытые коды	+	+	+	+	+
Обученные сети	+	–	+	+	+

встроенных предварительно обученных нейронных сетей. MXNet и Neon показывают хорошие результаты на тестах производительности [111, 121].

## Заключение

С момента возникновения нейронных сетей произошло много изменений в их архитектуре и методах обучения. В настоящее время доминирующими являются два типа архитектур: сверточные сети, которые успешно применяются для задач компьютерного зрения, и рекуррентные сети, активно используемые для задач обработки естественного языка.

Ранние сверточные сети обучались путем комбинации обучения с учителем и без учителя с использованием автокодировщиков и глубоких сетей доверия. Современные методы, такие как остаточное обучение, позволяют использовать только обучение с учителем и отказаться от предобучения, что ускоряет и упрощает процесс обучения. Также важным направлением в развитии сверточных нейронных сетей является передача обучения (transfer learning) [122]. Этот подход предполагает использование нейронных сетей, обученных на одних данных, для решения других типов задач. При этом применяется тонкая настройка сети и дообучение на данных от интересующей нас задачи. В результате сокращается время обучения и расширяется область применения предварительно обученных нейронных сетей. Перспективным также является совместное использование сверточных и рекуррентных нейронных сетей с обучением с подкреплением [123].

Для задач обработки естественного языка, и более общего случая обработки последовательностей, в настоящее время используются рекуррентные нейронные сети.

Среди них наиболее эффективными являются сети долго-краткосрочной памяти и управляемые рекуррентные нейронные сети, т.к. они позволяют запоминать интересные события на длительное время [124]. Дополнительным преимуществом рекуррентных сетей является возможность обучения без учителя и без предварительно размеченного набора данных.

Широкое распространение практического применения нейронных сетей является возможным благодаря наличию большого количества готовых решений для обучения глубоких нейронных сетей [109, 112–115, 117–120], в том числе с возможностью использования современных многоядерных процессоров, ускорителей вычислений GPU и Intel Xeon Phi, а также вычислительных кластеров с распределенной памятью.

*Работа поддержана проектом УрО РАН № 15-7-1-8.*

## Литература

1. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. 2015. Vol. 521. P. 436–444. DOI: 10.1038/nature14539.
2. Ravi D., Wong Ch., Deligianni F., et al. Deep Learning for Health Informatics // IEEE Journal of Biomedical and Health Informatics. 2017. Vol. 21, No. 1. P. 4–21. DOI: 10.1109/JBHI.2016.2636665.
3. Schmidhuber J. Deep Learning in Neural Networks: an Overview // Neural Networks. 2015. Vol. 1. P. 85–117, DOI: 10.1016/j.neunet.2014.09.003.
4. McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity // The Bulletin of Mathematical Biophysics. 1943. Vol. 5, No. 4. P. 115–133. DOI: 10.1007/BF02478259.
5. Hinton G., Salakhutdinov R. Reducing the Dimensionality of Data with Neural Networks // Science. 2006. Vol. 313, No. 5786. P. 504–507. DOI: 10.1126/science.1127647.
6. Hinton G.E., Osindero S., Teh Y.-W. A Fast Learning Algorithm for Deep Belief Nets // Neural Computing. 2006. Vol. 18, No. 7. P. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.
7. Sîma J. Loading Deep Networks Is Hard // Neural Computation. 1994. Vol. 6, No. 5. P. 842–850. DOI: 10.1162/neco.1994.6.5.842.
8. Windisch D. Loading Deep Networks Is Hard: The Pyramidal Case // Neural Computation. 2005. Vol. 17, No. 2. P. 487–502. DOI: 10.1162/0899766053011519.
9. Gomez F.J., Schmidhuber J. Co-Evolving Recurrent Neurons Learn Deep Memory POMDPs // Proc. of the 2005 Conference on Genetic and Evolutionary Computation (GECCO) (Washington, DC, USA, June 25–29, 2005), 2005. P. 491–498. DOI: 10.1145/1068009.1068092.
10. Ciresan D.C., Meier U., Gambardella L.M., Schmidhuber J. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition // Neural Computation. 2010. Vol. 22, No. 12. P. 3207–3220. DOI: 10.1162/NECO\_a\_00052.
11. He K., Zhang X., Ren S., et al. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (Las Vegas, NV, USA, 27–30 June 2016), 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.

12. Rumelhart D.E., Hinton G.E., McClelland J.L. A General Framework for Parallel Distributed Processing // *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1986. Vol. 1, P. 45–76. DOI: 10.1016/B978-1-4832-1446-7.50010-8.
13. LeCun Y., Bottou L., Orr G.B. Efficient BackProp // *Neural Networks: Tricks of the Trade*. 1998. P. 9–50. DOI: 10.1007/3-540-49430-8\_2.
14. Broomhead D.S., Lowe D. Multivariable Functional Interpolation and Adaptive Networks // *Complex Systems*. Vol. 2. P. 321–355. DOI: 10.1016/0167-6911(92)90025-N.
15. Stone M.N. The Generalized Weierstrass Approximation Theorem // *Mathematics Magazine*. 1948. Vol. 21, No. 4. P. 167–184. DOI: 10.2307/3029750.
16. Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. и др. *Нейроинформатика*. Новосибирск: Наука. 1998. С. 296.
17. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // *Neural Networks*. 1989. Vol. 2, No. 5. P. 359–366. DOI: 10.1016/0893-6080(89)90020-8.
18. Mhaskar H.N., Micchelli Ch.A. Approximation by Superposition of Sigmoidal and Radial Basis Functions // *Advances in Applied Mathematics*. 1992. Vol. 13, No. 13. P. 350–373. DOI: 10.1016/0196-8858(92)90016-P.
19. Hebb D.O. *The Organization of Behavior*. New York: Wiley. 1949. 335 p. DOI: 10.1016/S0361-9230(99)00182-3.
20. Novikoff A.B. On Convergence Proofs on Perceptrons // *Symposium on the Mathematical Theory of Automata*. 1962. Vol. 12. P. 615–622.
21. Rosenblatt F. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain // *Psychological Review*. 1958. P. 65–386. DOI: 10.1037/h0042519.
22. Widrow B., Hoff M. Associative Storage and Retrieval of Digital Information in Networks of Adaptive Neurons // *Biological Prototypes and Synthetic Systems*. 1962. Vol. 1. 160 p. DOI: 10.1007/978-1-4684-1716-6\_25.
23. Narendra K.S., Thathatchar M.A.L. Learning Automata – a Survey // *IEEE Transactions on Systems, Man, and Cybernetics*. 1974. Vol. 4. P. 323–334. DOI: 10.1109/tsmc.1974.5408453.
24. Rosenblatt F. *Principles of Neurodynamics; Perceptrons and the Theory of Brain Mechanisms*. 1962. Washington: Spartan Books. 616 p. DOI: 10.1007/978-3-642-70911-1\_20.
25. Grossberg S. Some Networks That Can Learn, Remember, and Reproduce any Number of Complicated Space-Time Patterns // *International Journal of Mathematics and Mechanics*. 1969. Vol. 19. P. 53–91. DOI: 10.1512/iumj.1970.19.19007.
26. Kohonen T. Correlation Matrix Memories // *IEEE Transactions on Computers*. 1972. Vol. 100, No. 4. P. 353–359. DOI: 10.1109/tc.1972.5008975.
27. von der Malsburg C. Self-Organization of Orientation Sensitive Cells in the Striate Cortex // *Kybernetik*. 1973. Vol. 14, No. 2. P. 85–100. DOI: 10.1007/bf00288907.
28. Willshaw D.J., von der Malsburg C. How Patterned Neural Connections Can Be Set Up by Self-Organization // *Proceedings of the Royal Society London B*. 1976. Vol. 194. P. 431–445. DOI: 10.1098/rspb.1976.0087.

29. Ivakhnenko A.G. Heuristic Self-Organization in Problems of Engineering Cybernetics // Automatica. 1970. Vol. 6, No. 2. P. 207–219. DOI: 10.1016/0005-1098(70)90092-0.
30. Ivakhnenko A.G. Polynomial Theory of Complex Systems // IEEE Transactions on Systems, Man and Cybernetics. 1971. Vol. 4. P. 364–378. DOI: 10.1109/tsmc.1971.4308320.
31. Ikeda S., Ochiai M., Sawaragi Y. Sequential GMDH Algorithm and Its Application to River Flow Prediction // IEEE Transactions on Systems, Man and Cybernetics. 1976. Vol. 7. P. 473–479. DOI: 10.1109/tsmc.1976.4309532.
32. Witczak M., Korbicz J., Mrugalski M., et al. A GMDH Neural Network-Based Approach to Robust Fault Diagnosis: Application to the DAMADICS Benchmark Problem // Control Engineering Practice. 2006. Vol. 14, No. 6. P. 671–683. DOI: 10.1016/j.conengprac.2005.04.007.
33. Kondo T., Ueno J. Multi-Layered GMDH-type Neural Network Self-Selecting Optimum Neural Network Architecture and Its Application to 3-Dimensional Medical Image Recognition of Blood Vessels // International Journal of Innovative Computing, Information and Control. 2008. Vol. 4, No. 1. P. 175–187.
34. Linnainmaa S. The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors. University of Helsinki. 1970.
35. Linnainmaa S. Taylor Expansion of the Accumulated Rounding Error // BIT Numerical Mathematics. 1976. Vol. 16, No. 2. P. 146–160. DOI: 10.1007/bf01931367.
36. Werbos P.J. Applications of Advances in Nonlinear Sensitivity Analysis // Lecture Notes in Control and Information Sciences. 1981. Vol. 38, P. 762–770. DOI: 10.1007/BFb0006203.
37. Parker D.B. Learning Logic. Technical Report TR-47, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA. 1985.
38. LeCun Y. A Theoretical Framework for Back-Propagation // Proceedings of the 1988 Connectionist Models Summer School (Pittsburgh, Pennsylvania, USA, June 17–26, 1988), 1988. P. 21–28.
39. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Internal Representations by Error Propagation // Parallel Distributed Processing. 1986. Vol. 1. P. 318–362. DOI: 10.1016/b978-1-4832-1446-7.50035-2.
40. Qian N. On the Momentum Term in Gradient Descent Learning Algorithms // Neural Networks: The Official Journal of the International Neural Network Society. 1999. Vol. 12, No. 1. P. 145–151. DOI: 10.1016/s0893-6080(98)00116-6.
41. Duchi J., Hazan E., Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization // Journal of Machine Learning Research. 2011. Vol. 12. P. 2121–2159.
42. Kingma D.P., Ba J.L. Adam: a Method for Stochastic Optimization // International Conference on Learning Representations (San Diego, USA, May 7-9, 2015), 2015. P. 1–13.
43. Fukushima K. Neocognitron: a Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position // Biological Cybernetics. 1980. Vol. 36, No. 4. P. 193–202. DOI: 10.1007/BF00344251.

44. Wiesel D.H., Hubel T.N. Receptive Fields of Single Neurones in the Cat's Striate Cortex // *The Journal of Physiology*. 1959. Vol. 148, No. 3. P. 574–591. DOI: 10.1113/jphysiol.1959.sp006308.
45. Fukushima K. Artificial Vision by Multi-Layered Neural Networks: Neocognitron and its Advances // *Neural Networks*. 2013. Vol. 37. P. 103–119. DOI: 10.1016/j.neunet.2012.09.016.
46. Fukushima K. Training Multi-Layered Neural Network Neocognitron // *Neural Networks*. 2013. Vol. 40. P. 18–31. DOI: 10.1016/j.neunet.2013.01.001.
47. Fukushima K. Increasing Robustness Against Background Noise: Visual Pattern Recognition by a Neocognitron // *Neural Networks*. 2011. Vol. 24, No. 7. P. 767–778. DOI: 10.1016/j.neunet.2011.03.017.
48. Ballard D.H. Modular Learning in Neural Networks // *Proceedings of the Sixth National Conference on Artificial Intelligence (Seattle, Washington, USA, July 13–17, 1987)*, 1987. Vol. 1. P. 279–284.
49. Hinton G.E., McClelland J.L. Learning Representations by Recirculation // *Neural Information Processing Systems*. 1998. American Institute of Physics. P. 358–366.
50. Wolpert D.H. Stacked Generalization // *Neural Networks*. 1992. Vol. 5, No. 2. P. 241–259. DOI: 10.1016/s0893-6080(05)80023-1.
51. Ting K.M., Witten I.H. Stacked Generalization: When Does It Work? // *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (Nagoya, Japan, August 23–29, 1997)*, 1997. P. 866–871.
52. LeCun Y., Boser B., Denker J.S., et al. Back-Propagation Applied to Handwritten Zip Code Recognition // *Neural Computation*. 1998. Vol. 1, No. 4. P. 541–551. DOI: 10.1162/neco.1989.1.4.541.
53. LeCun Y., Boser B., Denker J.S., et al. Handwritten Digit Recognition with a Back-Propagation Network // *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann. 1990. P. 396–404.
54. Baldi P., Chauvin Y. Neural Networks for Fingerprint Recognition // *Neural Computation*. 1993. Vol. 5, No. 3. P. 402–418. DOI: 10.1007/978-3-642-76153-9\_35.
55. Elman J.L. Finding Structure in Time // *Cognitive Science*. 1990. Vol. 14, No. 2. P. 179–211. DOI: 10.1207/s15516709cog1402\_1.
56. Jordan M.I. Serial Order: a Parallel Distributed Processing Approach. Institute for Cognitive Science, University of California, San Diego. ICS Report 8604. 1986. P. 40.
57. Jordan M.I. Serial Order: a Parallel Distributed Processing Approach // *Advances in Psychology*. 1997. Vol. 121. P. 471–495. DOI: 10.1016/s0166-4115(97)80111-2.
58. Hochreiter S. Untersuchungen zu Dynamischen Neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer. Technische Universität München, 1991.
59. Hochreiter S., Bengio Y., Frasconi P., et al. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies // *A Field Guide to Dynamical Recurrent Neural Networks*. Wiley-IEEE Press. 2001. P. 237–243. DOI: 10.1109/9780470544037.ch14.
60. Bengio Y., Simard P., Frasconi P. Learning Long-Term Dependencies with Gradient Descent is Difficult // *IEEE Transactions on Neural Networks*. 1994. Vol. 5, No. 2. P. 157–166. DOI: 10.1109/72.279181.

61. Tiño P., Hammer B. Architectural Bias in Recurrent Neural Networks: Fractal Analysis // *Neural Computation*. 2004. Vol. 15, No. 8. P. 1931–1957. DOI: 10.1162/08997660360675099.
62. Hochreiter S., Schmidhuber J. Bridging Long Time Lags by Weight Guessing and “Long Short-Term Memory” // *Spatiotemporal Models in Biological and Artificial Systems*. 1996. Vol. 37. P. 65–72.
63. Schmidhuber J., Wierstra D., Gagliolo M., et al. Training Recurrent Networks by Evolino. // *Neural Computation*. 2007. Vol. 19, No. 3. P. 757–779. DOI: 10.1162/neco.2007.19.3.757.
64. Levin L.A. Universal Sequential Search Problems // *Problems of Information Transmission*. 1997. Vol. 9, No. 3. P. 265–266.
65. Schmidhuber J. Discovering Neural Nets with Low Kolmogorov Complexity and High Generalization Capability // *Neural Networks*. 1997. Vol. 10, No. 5. P. 857–873. DOI: 10.1016/s0893-6080(96)00127-x.
66. Møller, M.F. Exact Calculation of the Product of the Hessian Matrix of Feed-Forward Network Error Functions and a Vector in O(N) Time. Computer Science Department, Aarhus University, Denmark. 1993. No. PB-432. DOI: 10.7146/dpb.v22i432.6748.
67. Pearlmutter B.A. Fast Exact Multiplication by the Hessian // *Neural Computation*. 1994. Vol. 6, No. 1. P. 147–160. DOI: 10.1162/neco.1994.6.1.147.
68. Schraudolph N.N. Fast Curvature Matrix-Vector Products for Second-Order Gradient Descent // *Neural Computation*. 2002. Vol. 14, No. 7. P. 1723–1738. DOI: 10.1162/08997660260028683.
69. Martens J. Deep Learning via Hessian-Free Optimization // *Proceedings of the 27th International Conference on Machine Learning (ICML-10) (Haifa, Israel, June 21–24, 2010)*, 2010. P. 735–742.
70. Martens J., Sutskever I. Learning Recurrent Neural Networks with Hessian-Free Optimization // *Proceedings of the 28th International Conference on Machine Learning (ICML-11) (Bellevue, Washington, USA, June 28 – July 02, 2011)*, 2011. P. 1033–1040.
71. Schmidhuber J. Learning Complex, Extended Sequences Using the Principle of History Compression // *Neural Computation*. 1992. Vol. 4, No. 2. P. 234–242. DOI: 10.1162/neco.1992.4.2.234.
72. Connor J., Martin D.R., Atlas L.E. Recurrent Neural Networks and Robust Time Series Prediction // *IEEE Transactions on Neural Networks*. 1994. Vol. 5, No. 2. P. 240–254. DOI: 10.1109/72.279188.
73. Dorffner G. Neural Networks for Time Series Processing // *Neural Network World*. 1996. Vol. 6. P. 447–468.
74. Schmidhuber J., Mozer M.C., Prelinger D. Continuous History Compression // *Proceedings of International Workshop on Neural Networks (Aachen, Germany, 1993)*, 1993. P. 87–95.
75. Hochreiter S., Schmidhuber J. Long Short-Term Memory // *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
76. Gers F.A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM // *Neural Computation*. 2000. Vol. 12, No. 10. P. 2451–2471. DOI: 10.1162/089976600300015015.

77. Pérez-Ortiz J.A., Gers F.A., Eck D., et al. Kalman Filters Improve LSTM Network Performance in Problems Unsolvable by Traditional Recurrent Nets // *Neural Networks*. 2003. Vol. 16, No. 2. P. 241–250. DOI: 10.1016/s0893-6080(02)00219-8.
78. Weng J., Ahuja N., Huang T.S. Cresceptron: a Self-Organizing Neural Network Which Grows Adaptively // *International Joint Conference on Neural Networks (IJCNN)* (Baltimore, MD, USA, 7–11 June 1992). 1992. Vol. 1. P. 576–581. DOI: 10.1109/ijcnn.1992.287150.
79. Weng J.J., Ahuja N., Huang T.S. Learning Recognition and Segmentation Using the Cresceptron // *International Journal of Computer Vision*. 1997. Vol. 25, No. 2. P. 109–143. DOI: 10.1023/a:1007967800668.
80. Ranzato M.A., Huang F.J., Boureau Y.L., et al. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition // *IEEE Conference on Computer Vision and Pattern Recognition (Minneapolis, MN, USA, 17–22 June 2007)*, 2007. P. 1–8. DOI: 10.1109/cvpr.2007.383157.
81. Scherer D., Müller A., Behnke S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition // *Lecture Notes in Computer Science*. 2010. Vol. 6354, P. 92–101. DOI: 10.1007/978-3-642-15825-4\_10.
82. Smolensky P. Information Processing in Dynamical Systems: Foundations of Harmony Theory // *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1986. Vol. 1. P. 194–281.
83. Hinton G.E., Sejnowski T.E. Learning and Relearning in Boltzmann Machines // *Parallel Distributed Processing*. 1986. Vol. 1. P. 282–317.
84. Memisevic R., Hinton G.E. Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines // *Neural Computation*. 2010. Vol. 22, No. 6. P. 1473–1492. DOI: 10.1162/neco.2010.01-09-953.
85. Mohamed A., Hinton G.E. Phone Recognition Using Restricted Boltzmann Machines // *IEEE International Conference on Acoustics, Speech and Signal Processing (Dallas, TX, USA, 14–19 March 2010)*, 2010. P. 4354–4357. DOI: 10.1109/icassp.2010.5495651.
86. Salakhutdinov R., Hinton G. Semantic Hashing // *International Journal of Approximate Reasoning*. 2009. Vol. 50, No. 7. P. 969–978. DOI: 10.1016/j.ijar.2008.11.006.
87. Bengio Y., Lamblin P., Popovici D., et al. Greedy Layer-Wise Training of Deep Networks // *Advances in Neural Information Processing Systems 19*. 2007. P. 153–160.
88. Vincent P., Hugo L., Bengio Y., et al. Extracting and Composing Robust Features with Denoising Autoencoders // *Proceedings of the 25th international Conference on Machine learning (Helsinki, Finland, July 05–09, 2008)*. 2008. P. 1096–1103. DOI: 10.1145/1390156.1390294.
89. Erhan D., Bengio Y., Courville A., et al. Why Does Unsupervised Pre-Training Help Deep Learning? // *Journal of Machine Learning Research*. 2010. Vol. 11. P. 625–660.
90. Arel I., Rose D.C., Karnowski T.P. Deep Machine Learning – a New Frontier in Artificial Intelligence Research // *Computational Intelligence Magazine, IEEE*. 2010. Vol. 5, No. 4. P. 13–18. DOI: 10.1109/mci.2010.938364.
91. Viren J., Sebastian S. Natural Image Denoising with Convolutional Networks // *Advances in Neural Information Processing Systems (NIPS) 21*. 2009. P. 769–776.

92. Razavian A.Sh., Azizpour H., Sullivan J., et al. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition // Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (Washington, DC, USA, June 23–28, 2014), 2014. P. 512–519. DOI: 10.1109/cvprw.2014.131.
93. Ruochen W., Zhe X. A Pedestrian and Vehicle Rapid Identification Model Based on Convolutional Neural Network // Proceedings of the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS '15) (Zhangjiajie, China, August 19–21, 2015), 2015. P. 32:1–32:4. DOI: 10.1145/2808492.2808524.
94. Boominathan L., Kruthiventi S.S., Babu R.V. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting // Proceedings of the 2016 ACM on Multimedia Conference (Amsterdam, The Netherlands, October 15–19, 2016), 2016. P. 640–644. DOI: 10.1145/2964284.2967300.
95. Kinnikar A., Husain M., Meena S.M. Face Recognition Using Gabor Filter And Convolutional Neural Network // Proceedings of the International Conference on Informatics and Analytics (Pondicherry, India, August 25–26, 2016), 2016. P. 113:1–113:4. DOI: 10.1145/2980258.2982104.
96. Hahnloser R.H.R., Sarpeshkar R., Mahowald M.A., et al. Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit // Nature. 2000. Vol. 405. P. 947–951. DOI: 10.1038/35016072.
97. Hahnloser R.H.R., Seung H.S., Slotine J.J. Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks // Neural Computation. 2003. Vol. 15, No. 3. P. 621–638. DOI: 10.1162/089976603321192103.
98. Glorot X., Bordes A., Bengio Y. Deep Sparse Rectifier Neural Networks // Journal of Machine Learning Research. 2011. Vol. 15. P. 315–323.
99. Glorot X., Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks // Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10) (Sardinia, Italy, May 13–15, 2010). Society for Artificial Intelligence and Statistics. 2010. P. 249–256.
100. He K., Zhang X., Ren Sh. et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification // Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (Santiago, Chile, December 7–13, 2015), 2015. P. 1026–1034. DOI: 10.1109/ICCV.2015.123.
101. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // JMLR Workshop and Conference Proceedings. Proceedings of the 32nd International Conference on Machine Learning (Lille, France, July 06–11, 2015), 2015. Vol. 37. P. 448–456.
102. Szegedy C., Liu W, Jia Y. et al. Going Deeper with Convolutions // IEEE Conference on Computer Vision and Pattern Recognition (Boston, MA, USA, June 7–12, 2015), 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594.
103. Szegedy C., Vanhoucke V., Ioffe S., et al. Rethinking the Inception Architecture for Computer Vision // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Seattle, WA, USA, Jun 27–30, 2016), 2016. P. 2818–2826. DOI: 10.1109/cvpr.2016.308.



104. Szegedy C., Ioffe S., Vanhoucke V., et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning // Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) (San Francisco, California, USA, February 4–9, 2017), 2017. P. 4278–4284.
105. Cho K., van Merriënboer B., Gulcehre C., et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar, October 25–29, 2014), 2014. P. 1724–1734. DOI: 10.3115/v1/d14-1179.
106. Cho K., van Merriënboer B., Bahdanau D., et al. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches // Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (Doha, Qatar, October 25, 2014), 2014. P. 103–111. DOI: 10.3115/v1/w14-4012.
107. Chung, J., Gulcehre, C., Cho, K., et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling // NIPS 2014 Workshop on Deep Learning (Montreal, Canada, December 12, 2014), 2014. P. 1–9.
108. He K., Sun J. Convolutional Neural Networks at Constrained Time Cost // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Boston, MA, USA, June 07–12, 2015), 2015. P. 5353–5360. DOI: 10.1109/CVPR.2015.7299173.
109. Jia Y., Shelhamer E., Donahue J., et al. Caffe: Convolutional Architecture for Fast Feature Embedding // Proceedings of the 22nd ACM International Conference on Multimedia (Orlando, FL, USA, November 03–07, 2014), 2014. P. 675–678. DOI: 10.1145/2647868.2654889
110. Kruchinin D., Dolotov E., Korniyakov K. et al. Comparison of Deep Learning Libraries on the Problem of Handwritten Digit Classification // Analysis of Images, Social Networks and Texts. Communications in Computer and Information Science. 2015. Vol. 542. P. 399–411. DOI: 10.1007/978-3-319-26123-2\_38.
111. Bahrampour S., Ramakrishnan N., Schott L., et al. Comparative Study of Deep Learning Software Frameworks. URL: <https://arxiv.org/abs/1511.06435> (дата обращения: 02.07.2017).
112. Bergstra J., Breuleux O., Bastien F., et al. Theano: a CPU and GPU Math Expression Compiler // Proceedings of the Python for Scientific Computing Conference (SciPy) (Austin, TX, USA, June 28 – July 3, 2010), 2010. P. 3–10.
113. Abadi M., Agarwal A. Barham P. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems // Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) (Savannah, GA, USA, November, 2–4, 2016), 2016. P. 265–283.
114. Collobert R., Kavukcuoglu K., Farabet C. Torch7: a Matlab-like Environment for Machine Learning // BigLearn, NIPS Workshop (Granada, Spain, December 12–17, 2011), 2011.
115. Seide F., Agarwal A. CNTK: Microsoft’s Open-Source Deep-Learning Toolkit // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16) (San Francisco, California, USA, August 13–17, 2016), 2016. P. 2135–2135. DOI: 10.1145/2939672.2945397.
116. Viebke A., Pillana S. The Potential of the Intel(r) Xeon Phi for Supervised Deep Learning // IEEE 17th International Conference on High Performance Computing and

- Communications (HPCC) (New York, USA, August 24–26, 2015), 2015. P. 758–765. DOI: 10.1109/hpcc-css-icess.2015.45.
117. Chollet. F., et al. Keras. 2015. URL: <https://github.com/fchollet/keras> (дата обращения: 02.07.2017).
118. PaddlePaddle: PArallel Distributed Deep LEarning. URL: <http://www.paddlepaddle.org/> (дата обращения: 02.07.2017).
119. Chen T., Li M., Li Y. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. URL: <https://arxiv.org/abs/1512.01274> (дата обращения: 02.07.2017).
120. Intel Nervana Reference Deep Learning Framework Committed to Best Performance on all Hardware. URL: <https://www.intelnervana.com/neon/> (дата обращения: 02.07.2017).
121. Shi Sh., Wang Q., Xu P. Benchmarking State-of-the-Art Deep Learning Software Tools. URL: <https://arxiv.org/abs/1608.07249> (дата обращения: 02.07.2017).
122. Weiss K., Khoshgoftaar T.M., Wang D. A Survey of Transfer Learning // Journal of Big Data. 2016. Vol. 3, No. 1. P. 1–9. DOI: 10.1186/s40537-016-0043-6
123. Ba J., Mnih V., Kavukcuoglu K. Multiple Object Recognition with Visual Attention // Proceedings of the International Conference on Learning Representations (ICLR) (San Diego, USA, May 7–9, 2015), 2015. P. 1–10.
124. Graves A., Mohamed A.R., Hinton G. Speech Recognition with Deep Recurrent Neural Networks // IEEE International Conference on Acoustics, Speech and Signal Processing (Vancouver, Canada, May 26–31, 2013), 2013. P. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.

Созыкин Андрей Владимирович, к.т.н., зав. отделом вычислительной техники, Институт математики и механики им. Н.Н.Красовского УрО РАН, зав. кафедрой, высокопроизводительных компьютерных технологий, Уральский федеральный университет (Екатеринбург, Российская Федерация)

---

DOI: 10.14529/cmse170303

## AN OVERVIEW OF METHODS FOR DEEP LEARNING IN NEURAL NETWORKS

© 2017 A.V. Sozykin

*N.N. Krasovskii Institute of Mathematics and Mechanics (S.Kovalevskaya str. 16,  
Yekaterinburg, 620990 Russia),*

*Ural Federal University (Mira str. 19, Yekaterinburg, 620002 Russia)*

*E-mail: avs@imm.uran.ru*

Received: 12.04.2017

At present, deep learning is becoming one of the most popular approach to creation of the artificial intelligences systems such as speech recognition, natural language processing, computer vision and so on. The paper presents a historical overview of deep learning in neural networks. The model of the artificial neural network is described as well as the learning algorithms for neural networks including the error backpropagation algorithm,

which is used to train deep neural networks. The development of neural networks architectures is presented including neocognitron, autoencoders, convolutional neural networks, restricted Boltzmann machine, deep belief networks, long short-term memory, gated recurrent neural networks, and residual networks. Training deep neural networks with many hidden layers is impeded by the vanishing gradient problem. The paper describes the approaches to solve this problem that provide the ability to train neural networks with more than hundred layers. An overview of popular deep learning libraries is presented. Nowadays, for computer vision tasks convolutional neural networks are utilized, while for sequence processing, including natural language processing, recurrent networks are preferred solution, primarily long short-term memory networks and gated recurrent neural networks.

*Keywords: deep learning, neural networks, machine learning.*

## FOR CITATION

Sozykin A.V. An Overview of Methods for Deep Learning in Neural Networks.. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 28–59. (in Russian) DOI: 10.14529/cmse170303.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. LeCun Y., Bengio Y., Hinton G. Deep Learning. *Nature*. 2015. vol. 521. pp. 436–444. DOI: 10.1038/nature14539.
2. Ravì D., Wong Ch., Deligianni F., et al. Deep Learning for Health Informatics. *IEEE Journal of Biomedical and Health Informatics*. 2017. vol. 21, no. 1. pp. 4–21. DOI: 10.1109/JBHI.2016.2636665.
3. Schmidhuber J. Deep Learning in Neural Networks: an Overview. *Neural Networks*. 2015. vol. 1. pp. 85–117, DOI: 10.1016/j.neunet.2014.09.003.
4. McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*. 1943. vol. 5, no. 4. pp. 115–133. DOI: 10.1007/BF02478259.
5. Hinton G., Salakhutdinov R. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006. vol. 313, no. 5786. pp. 504–507. DOI: 10.1126/science.1127647.
6. Hinton G.E., Osindero S., Teh Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computing*. 2006. vol. 18, no. 7. pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.
7. Sîma J. Loading Deep Networks Is Hard. *Neural Computation*. 1994. vol. 6, no. 5. pp. 842–850. DOI: 10.1162/neco.1994.6.5.842.
8. Windisch D. Loading Deep Networks Is Hard: The Pyramidal Case. *Neural Computation*. 2005. vol. 17, no. 2. pp. 487–502. DOI: 10.1162/0899766053011519.
9. Gomez F.J., Schmidhuber J. Co-Evolving Recurrent Neurons Learn Deep Memory POMDPs. *Proc. of the 2005 Conference on Genetic and Evolutionary Computation (GECCO) (Washington, DC, USA, June 25–29, 2005)*, 2005. pp. 491–498. DOI: 10.1145/1068009.1068092.
10. Ciresan D.C., Meier U., Gambardella L.M., Schmidhuber J. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation*. 2010. vol. 22, no. 12. pp. 3207–3220. DOI: 10.1162/NECO\_a\_00052.

11. He K., Zhang X., Ren S., et al. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (Las Vegas, NV, USA, 27–30 June 2016), 2016. pp. 770–778. DOI: 10.1109/CVPR.2016.90.
12. Rumelhart D.E., Hinton G.E., McClelland J.L. A General Framework for Parallel Distributed Processing. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1986. vol. 1, pp. 45–76. DOI: 10.1016/B978-1-4832-1446-7.50010-8.
13. LeCun Y., Bottou L., Orr G.B. Efficient BackProp. *Neural Networks: Tricks of the Trade*. 1998. pp. 9–50. DOI: 10.1007/3-540-49430-8\_2.
14. Broomhead D.S., Lowe D. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*. vol. 2. pp. 321–355. DOI: 10.1016/0167-6911(92)90025-N.
15. Stone M.N. The Generalized Weierstrass Approximation Theorem. *Mathematics Magazine*. 1948. vol. 21, no. 4. pp. 167–184. DOI: 10.2307/3029750.
16. Gorban A.N., Dudin-Barkovsky V.L., Kiridin A.N., et al. *Nejroinformatika [Neuroinformatics]*. Novosibirsk, Science, 1998. 296 p.
17. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*. 1989. vol. 2, no. 5. pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.
18. Mhaskar H.N., Micchelli Ch.A. Approximation by Superposition of Sigmoidal and Radial Basis Functions. *Advances in Applied Mathematics*. 1992. vol. 13, no. 13. pp. 350–373. DOI: 10.1016/0196-8858(92)90016-P.
19. Hebb D.O. *The Organization of Behavior*. New York: Wiley. 1949. 335 p. DOI: 10.1016/S0361-9230(99)00182-3.
20. Novikoff A.B. On Convergence Proofs on Perceptrons. *Symposium on the Mathematical Theory of Automata*. 1962. vol. 12. pp. 615–622.
21. Rosenblatt F. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*. 1958. pp. 65–386. DOI: 10.1037/h0042519.
22. Widrow B., Hoff M. Associative Storage and Retrieval of Digital Information in Networks of Adaptive Neurons. *Biological Prototypes and Synthetic Systems*. 1962. vol. 1. 160 p. DOI: 10.1007/978-1-4684-1716-6\_25.
23. Narendra K.S., Thathatchar M.A.L. Learning Automata – a Survey. *IEEE Transactions on Systems, Man, and Cybernetics*. 1974. vol. 4. pp. 323–334. DOI: 10.1109/tsmc.1974.5408453.
24. Rosenblatt F. *Principles of Neurodynamics; Perceptrons and the Theory of Brain Mechanisms*. 1962. Washington: Spartan Books. 616 p. DOI: 10.1007/978-3-642-70911-1\_20.
25. Grossberg S. Some Networks That Can Learn, Remember, and Reproduce any Number of Complicated Space-Time Patterns. *International Journal of Mathematics and Mechanics*. 1969. vol. 19. pp. 53–91. DOI: 10.1512/iumj.1970.19.19007.
26. Kohonen T. Correlation Matrix Memories. *IEEE Transactions on Computers*. 1972. vol. 100, no. 4. pp. 353–359. DOI: 10.1109/tc.1972.5008975.
27. von der Malsburg C. Self-Organization of Orientation Sensitive Cells in the Striate Cortex. *Kybernetik*. 1973. vol. 14, no. 2. pp. 85–100. DOI: 10.1007/bf00288907.

28. Willshaw D.J., von der Malsburg C. How Patterned Neural Connections Can Be Set Up by Self-Organization. *Proceedings of the Royal Society London B*. 1976. vol. 194. pp. 431–445. DOI: 10.1098/rspb.1976.0087.
29. Ivakhnenko A.G. Heuristic Self-Organization in Problems of Engineering Cybernetics. *Automatica*. 1970. vol. 6, no. 2. pp. 207–219. DOI: 10.1016/0005-1098(70)90092-0.
30. Ivakhnenko A.G. Polynomial Theory of Complex Systems. *IEEE Transactions on Systems, Man and Cybernetics*. 1971. vol. 4. pp. 364–378. DOI: 10.1109/tsmc.1971.4308320.
31. Ikeda S., Ochiai M., Sawaragi Y. Sequential GMDH Algorithm and Its Application to River Flow Prediction. *IEEE Transactions on Systems, Man and Cybernetics*. 1976. vol. 7. pp. 473–479. DOI: 10.1109/tsmc.1976.4309532.
32. Witczak M, Korbicz J, Mrugalski M., et al. A GMDH Neural Network-Based Approach to Robust Fault Diagnosis: Application to the DAMADICS Benchmark Problem. *Control Engineering Practice*. 2006. vol. 14, no. 6. pp. 671–683. DOI: 10.1016/j.conengprac.2005.04.007.
33. Kondo T., Ueno J. Multi-Layered GMDH-type Neural Network Self-Selecting Optimum Neural Network Architecture and Its Application to 3-Dimensional Medical Image Recognition of Blood Vessels. *International Journal of Innovative Computing, Information and Control*. 2008. vol. 4, no. 1. pp. 175–187.
34. Linnainmaa S. The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors. University of Helsinki. 1970.
35. Linnainmaa S. Taylor Expansion of the Accumulated Rounding Error. *BIT Numerical Mathematics*. 1976. vol. 16, no. 2. pp. 146–160. DOI: 10.1007/bf01931367.
36. Werbos P.J. Applications of Advances in Nonlinear Sensitivity Analysis. *Lecture Notes in Control and Information Sciences*. 1981. vol. 38, pp. 762–770. DOI: 10.1007/BFb0006203.
37. Parker D.B. Learning Logic. Technical Report TR-47, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA. 1985.
38. LeCun Y. A Theoretical Framework for Back-Propagation. *Proceedings of the 1988 Connectionist Models Summer School (Pittsburgh, Pennsylvania, USA, June 17–26, 1988)*, 1988. P. 21–28.
39. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing*. 1986. vol. 1. pp. 318–362. DOI: 10.1016/b978-1-4832-1446-7.50035-2.
40. Qian N. On the Momentum Term in Gradient Descent Learning Algorithms. *Neural Networks: The Official Journal of the International Neural Network Society*. 1999. vol. 12, no. 1. pp. 145–151. DOI: 10.1016/s0893-6080(98)00116-6.
41. Duchi J., Hazan E., Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*. 2011. vol. 12. pp. 2121–2159.
42. Kingma D.P., Ba J.L. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations (San Diego, USA, May 7-9, 2015)*, 2015. pp. 1–13.

43. Fukushima K. Neocognitron: a Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*. 1980. vol. 36, no. 4. pp. 193–202. DOI: 10.1007/BF00344251.
44. Wiesel D.H., Hubel T.N. Receptive Fields of Single Neurones in the Cat's Striate Cortex. *The Journal of Physiology*. 1959. vol. 148, no. 3. pp. 574–591. DOI: 10.1113/jphysiol.1959.sp006308.
45. Fukushima K. Artificial Vision by Multi-Layered Neural Networks: Neocognitron and its Advances. *Neural Networks*. 2013. vol. 37. pp. 103–119. DOI: 10.1016/j.neunet.2012.09.016.
46. Fukushima K. Training Multi-Layered Neural Network Neocognitron. *Neural Networks*. 2013. vol. 40. pp. 18–31. DOI: 10.1016/j.neunet.2013.01.001.
47. Fukushima K. Increasing Robustness Against Background Noise: Visual Pattern Recognition by a Neocognitron. *Neural Networks*. 2011. vol. 24, no. 7. pp. 767–778. DOI: 10.1016/j.neunet.2011.03.017.
48. Ballard D.H. Modular Learning in Neural Networks. *Proceedings of the Sixth National Conference on Artificial Intelligence (Seattle, Washington, USA, July 13–17, 1987)*, 1987. vol. 1. pp. 279–284.
49. Hinton G.E., McClelland J.L. Learning Representations by Recirculation. *Neural Information Processing Systems*. 1998. American Institute of Physics. pp. 358–366.
50. Wolpert D.H. Stacked Generalization. *Neural Networks*. 1992. vol. 5, no. 2. pp. 241–259. DOI: 10.1016/s0893-6080(05)80023-1.
51. Ting K.M., Witten I.H. Stacked Generalization: When Does It Work? *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (Nagoya, Japan, August 23–29, 1997)*, 1997. pp. 866–871.
52. LeCun Y., Boser B., Denker J.S., et al. Back-Propagation Applied to Handwritten Zip Code Recognition. *Neural Computation*. 1998. vol. 1, no. 4. pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.
53. LeCun Y., Boser B., Denker J.S., et al. Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann. 1990. pp. 396–404.
54. Baldi P., Chauvin Y. Neural Networks for Fingerprint Recognition. *Neural Computation*. 1993. vol. 5, no. 3. pp. 402–418. DOI: 10.1007/978-3-642-76153-9\_35.
55. Elman J.L. Finding Structure in Time. *Cognitive Science*. 1990. vol. 14, no. 2. pp. 179–211. DOI: 10.1207/s15516709cog1402\_1.
56. Jordan M.I. Serial Order: a Parallel Distributed Processing Approach. *Institute for Cognitive Science, University of California, San Diego. ICS Report 8604*. 1986. pp. 40.
57. Jordan M.I. Serial Order: a Parallel Distributed Processing Approach. *Advances in Psychology*. 1997. vol. 121. pp. 471–495. DOI: 10.1016/s0166-4115(97)80111-2.
58. Hochreiter S. Untersuchungen zu Dynamischen Neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer. Technische Universität München, 1991.
59. Hochreiter S., Bengio Y., Frasconi P., et al. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*. Wiley-IEEE Press. 2001. pp. 237–243. DOI: 10.1109/9780470544037.ch14.

60. Bengio Y., Simard P., Frasconi P. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*. 1994. vol. 5, no. 2. pp. 157–166. DOI: 10.1109/72.279181.
61. Tiño P., Hammer B. Architectural Bias in Recurrent Neural Networks: Fractal Analysis. *Neural Computation*. 2004. vol. 15, no. 8. pp. 1931–1957. DOI: 10.1162/08997660360675099.
62. Hochreiter S., Schmidhuber J. Bridging Long Time Lags by Weight Guessing and “Long Short-Term Memory”. *Spatiotemporal Models in Biological and Artificial Systems*. 1996. vol. 37. pp. 65–72.
63. Schmidhuber J., Wierstra D., Gagliolo M., et al. Training Recurrent Networks by Evolino.. *Neural Computation*. 2007. vol. 19, no. 3. pp. 757–779. DOI: 10.1162/neco.2007.19.3.757.
64. Levin L.A. Universal Sequential Search Problems. *Problems of Information Transmission*. 1997. vol. 9, no. 3. pp. 265–266.
65. Schmidhuber J. Discovering Neural Nets with Low Kolmogorov Complexity and High Generalization Capability. *Neural Networks*. 1997. vol. 10, no. 5. pp. 857–873. DOI: 10.1016/s0893-6080(96)00127-x.
66. Møller, M.F. Exact Calculation of the Product of the Hessian Matrix of Feed-Forward Network Error Functions and a Vector in  $O(N)$  Time. Computer Science Department, Aarhus University, Denmark. 1993. no. PB-432. DOI: 10.7146/dpb.v22i432.6748.
67. Pearlmutter B.A. Fast Exact Multiplication by the Hessian. *Neural Computation*. 1994. vol. 6, no. 1. pp. 147–160. DOI: 10.1162/neco.1994.6.1.147.
68. Schraudolph N.N. Fast Curvature Matrix-Vector Products for Second-Order Gradient Descent. *Neural Computation*. 2002. vol. 14, no. 7. pp. 1723–1738. DOI: 10.1162/08997660260028683.
69. Martens J. Deep Learning via Hessian-Free Optimization. *Proceedings of the 27th International Conference on Machine Learning (ICML-10) (Haifa, Israel, June 21–24, 2010)*, 2010. pp. 735–742.
70. Martens J., Sutskever I. Learning Recurrent Neural Networks with Hessian-Free Optimization. *Proceedings of the 28th International Conference on Machine Learning (ICML-11) (Bellevue, Washington, USA, June 28 – July 02, 2011)*, 2011. pp. 1033–1040.
71. Schmidhuber J. Learning Complex, Extended Sequences Using the Principle of History Compression. *Neural Computation*. 1992. vol. 4, no. 2. pp. 234–242. DOI: 10.1162/neco.1992.4.2.234.
72. Connor J., Martin D.R., Atlas L.E. Recurrent Neural Networks and Robust Time Series Prediction. *IEEE Transactions on Neural Networks*. 1994. vol. 5, no. 2. pp. 240–254. DOI: 10.1109/72.279188.
73. Dorffner G. Neural Networks for Time Series Processing. *Neural Network World*. 1996. vol. 6. pp. 447–468.
74. Schmidhuber J., Mozer M.C., Prelinger D. Continuous History Compression. *Proceedings of International Workshop on Neural Networks (Aachen, Germany, 1993)*, 1993. pp. 87–95.
75. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. vol. 9, no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

76. Gers F.A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*. 2000. vol. 12, no. 10. pp. 2451–2471. DOI: 10.1162/089976600300015015.
77. Pérez-Ortiz J.A., Gers F.A., Eck D., et al. Kalman Filters Improve LSTM Network Performance in Problems Unsolvable by Traditional Recurrent Nets. *Neural Networks*. 2003. vol. 16, no. 2. pp. 241–250. DOI: 10.1016/s0893-6080(02)00219-8.
78. Weng J., Ahuja N., Huang T.S. Cresceptron: a Self-Organizing Neural Network Which Grows Adaptively. *International Joint Conference on Neural Networks (IJCNN)* (Baltimore, MD, USA, 7–11 June 1992). 1992. vol. 1. pp. 576–581. DOI: 10.1109/ijcnn.1992.287150.
79. Weng J.J., Ahuja N., Huang T.S. Learning Recognition and Segmentation Using the Cresceptron. *International Journal of Computer Vision*. 1997. vol. 25, no. 2. pp. 109–143. DOI: 10.1023/a:1007967800668.
80. Ranzato M.A., Huang F.J., Boureau Y.L., et al. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (Minneapolis, MN, USA, 17–22 June 2007)*, 2007. pp. 1–8. DOI: 10.1109/cvpr.2007.383157.
81. Scherer D., Müller A., Behnke S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. *Lecture Notes in Computer Science*. 2010. vol. 6354, pp. 92–101. DOI: 10.1007/978-3-642-15825-4\_10.
82. Smolensky P. Information Processing in Dynamical Systems: Foundations of Harmony Theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1986. vol. 1. pp. 194–281.
83. Hinton G.E., Sejnowski T.E. Learning and Relearning in Boltzmann Machines. *Parallel Distributed Processing*. 1986. vol. 1. pp. 282–317.
84. Memisevic R., Hinton G.E. Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines. *Neural Computation*. 2010. vol. 22, no. 6. pp. 1473–1492. DOI: 10.1162/neco.2010.01-09-953.
85. Mohamed A., Hinton G.E. Phone Recognition Using Restricted Boltzmann Machines. *IEEE International Conference on Acoustics, Speech and Signal Processing (Dallas, TX, USA, 14–19 March 2010)*, 2010. pp. 4354–4357. DOI: 10.1109/icassp.2010.5495651.
86. Salakhutdinov R., Hinton G. Semantic Hashing. *International Journal of Approximate Reasoning*. 2009. vol. 50, no. 7. pp. 969–978. DOI: 10.1016/j.ijar.2008.11.006.
87. Bengio Y., Lamblin P., Popovici D., et al. Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems 19*. 2007. pp. 153–160.
88. Vincent P., Hugo L., Bengio Y., et al. Extracting and Composing Robust Features with Denoising Autoencoders. *Proceedings of the 25th international Conference on Machine learning (Helsinki, Finland, July 05–09, 2008)*. 2008. pp. 1096–1103. DOI: 10.1145/1390156.1390294.
89. Erhan D., Bengio Y., Courville A., et al. Why Does Unsupervised Pre-Training Help Deep Learning?. *Journal of Machine Learning Research*. 2010. vol. 11. pp. 625–660.



90. Arel I., Rose D.C., Karnowski T.P. Deep Machine Learning – a New Frontier in Artificial Intelligence Research. *Computational Intelligence Magazine, IEEE*. 2010. vol. 5, no. 4. pp. 13–18. DOI: 10.1109/mci.2010.938364.
91. Viren J., Sebastian S. Natural Image Denoising with Convolutional Networks. *Advances in Neural Information Processing Systems (NIPS) 21*. 2009. pp. 769–776.
92. Razavian A.Sh., Azizpour H., Sullivan J., et al. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (Washington, DC, USA, June 23–28, 2014)*, 2014. pp. 512–519. DOI: 10.1109/cvprw.2014.131.
93. Ruochen W., Zhe X. A Pedestrian and Vehicle Rapid Identification Model Based on Convolutional Neural Network. *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS '15) (Zhangjiajie, China, August 19–21, 2015)*, 2015. pp. 32:1–32:4. DOI: 10.1145/2808492.2808524.
94. Boominathan L., Kruthiventi S.S., Babu R.V. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting. *Proceedings of the 2016 ACM on Multimedia Conference (Amsterdam, The Netherlands, October 15–19, 2016)*, 2016. pp. 640–644. DOI: 10.1145/2964284.2967300.
95. Kinnikar A., Husain M., Meena S.M. Face Recognition Using Gabor Filter And Convolutional Neural Network. *Proceedings of the International Conference on Informatics and Analytics (Pondicherry, India, August 25–26, 2016)*, 2016. pp. 113:1–113:4. DOI: 10.1145/2980258.2982104.
96. Hahnloser R.H.R., Sarpeshkar R., Mahowald M.A., et al. Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit. *Nature*. 2000. vol. 405. pp. 947–951. DOI: 10.1038/35016072.
97. Hahnloser R.H.R., Seung H.S., Slotine J.J. Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks. *Neural Computation*. 2003. vol. 15, no. 3. pp. 621–638. DOI: 10.1162/089976603321192103.
98. Glorot X., Bordes A., Bengio Y. Deep Sparse Rectifier Neural Networks. *Journal of Machine Learning Research*. 2011. vol. 15. pp. 315–323.
99. Glorot X., Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10) (Sardinia, Italy, May 13–15, 2010)*. Society for Artificial Intelligence and Statistics. 2010. pp. 249–256.
100. He K., Zhang X., Ren Sh. et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (Santiago, Chile, December 7–13, 2015)*, 2015. pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.
101. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *JMLR Workshop and Conference Proceedings. Proceedings of the 32nd International Conference on Machine Learning (Lille, France, July 06–11, 2015)*, 2015. vol. 37. pp. 448–456.

102. Szegedy C., Liu W, Jia Y. et al. Going Deeper with Convolutions. IEEE Conference on Computer Vision and Pattern Recognition (Boston, MA, USA, June 7–12, 2015), 2015. pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.
103. Szegedy C., Vanhoucke V., Ioffe S., et al. Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Seattle, WA, USA, Jun 27–30, 2016), 2016. pp. 2818–2826. DOI: 10.1109/cvpr.2016.308.
104. Szegedy C., Ioffe S., Vanhoucke V., et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) (San Francisco, California, USA, February 4–9, 2017), 2017. pp. 4278–4284.
105. Cho K., van Merriënboer B., Gulcehre C., et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar, October 25–29, 2014), 2014. pp. 1724–1734. DOI: 10.3115/v1/d14-1179.
106. Cho K., van Merriënboer B., Bahdanau D., et al. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (Doha, Qatar, October 25, 2014), 2014. pp. 103–111. DOI: 10.3115/v1/w14-4012.
107. Chung, J., Gulcehre, C., Cho, K., et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Workshop on Deep Learning (Montreal, Canada, December 12, 2014), 2014. pp. 1–9.
108. He K., Sun J. Convolutional Neural Networks at Constrained Time Cost. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Boston, MA, USA, June 07–12, 2015), 2015. pp. 5353–5360. DOI: 10.1109/CVPR.2015.7299173.
109. Jia Y., Shelhamer E., Donahue J., et al. Caffe: Convolutional Architecture for Fast Feature Embedding. Proceedings of the 22nd ACM International Conference on Multimedia (Orlando, FL, USA, November 03–07, 2014), 2014. pp. 675–678. DOI: 10.1145/2647868.2654889.
110. Kruchinin D., Dolotov E., Korniyakov K. et al. Comparison of Deep Learning Libraries on the Problem of Handwritten Digit Classification. Analysis of Images, Social Networks and Texts. Communications in Computer and Information Science. 2015. vol. 542. pp. 399–411. DOI: 10.1007/978-3-319-26123-2\_38.
111. Bahrampour S., Ramakrishnan N., Schott L., et al. Comparative Study of Deep Learning Software Frameworks. Available at: <https://arxiv.org/abs/1511.06435> (accessed: 02.07.2017).
112. Bergstra J., Breuleux O., Bastien F., et al. Theano: a CPU and GPU Math Expression Compiler. Proceedings of the Python for Scientific Computing Conference (SciPy) (Austin, TX, USA, June 28 – July 3, 2010), 2010. pp. 3–10.
113. Abadi M., Agarwal A. Barham P. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) (Savannah, GA, USA, November, 2–4, 2016), 2016. pp. 265–283.

114. Collobert R., Kavukcuoglu K., Farabet C. Torch7: a Matlab-like Environment for Machine Learning. BigLearn, NIPS Workshop (Granada, Spain, December 12–17, 2011), 2011.
115. Seide F., Agarwal A. CNTK: Microsoft’s Open-Source Deep-Learning Toolkit. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16) (San Francisco, California, USA, August 13–17, 2016), 2016. pp. 2135–2135. DOI: 10.1145/2939672.2945397.
116. Viebke A., Pllana S. The Potential of the Intel(r) Xeon Phi for Supervised Deep Learning. IEEE 17th International Conference on High Performance Computing and Communications (HPCC) (New York, USA, August 24–26, 2015), 2015. pp. 758–765. DOI: 10.1109/hpcc-css-icess.2015.45.
117. Chollet. F., et al. Keras. 2015. Available at: <https://github.com/fchollet/keras> (accessed: 02.07.2017).
118. PadlePadle: PArallel Distributed Deep LEarning. Available at: <http://www.paddlepaddle.org/> (accessed: 02.07.2017).
119. Chen T., Li M., Li Y. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. Available at: <https://arxiv.org/abs/1512.01274> (accessed: 02.07.2017).
120. Intel Nervana Reference Deep Learning Framework Committed to Best Performance on all Hardware. Available at: <https://www.intelnervana.com/neon/> (accessed: 02.07.2017).
121. Shi Sh., Wang Q., Xu P. Benchmarking State-of-the-Art Deep Learning Software Tools. Available at: <https://arxiv.org/abs/1608.07249> (accessed: 02.07.2017).
122. Weiss K., Khoshgoftaar T.M., Wang D. A Survey of Transfer Learning. Journal of Big Data. 2016. vol. 3, no. 1. pp. 1–9. DOI: 10.1186/s40537-016-0043-6.
123. Ba J., Mnih V., Kavukcuoglu K. Multiple Object Recognition with Visual Attention. Proceedings of the International Conference on Learning Representations (ICLR) (San Diego, USA, May 7–9, 2015), 2015. pp. 1–10.
124. Graves A., Mohamed A.R., Hinton G. Speech Recognition with Deep Recurrent Neural Networks. IEEE International Conference on Acoustics, Speech and Signal Processing (Vancouver, Canada, May 26–31, 2013), 2013. pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.

# АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ НЕЧЕТКИХ ПРАВИЛ ДЛЯ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ С ГУСЕНИЧНЫМ ШАССИ НА ОСНОВЕ ЧИСЛОВЫХ ДАННЫХ

© 2017 Е.А. Пташко, В.И. Ухоботов

Челябинский государственный университет  
(454001 Челябинск, ул. Братъев Каширинных, д. 129)

E-mail: eugene.ptashko@gmail.com, ukh@csu.ru

Поступила в редакцию: 08.06.2017

В данной работе рассматривается актуальная на данный момент проблема генерации набора нечетких правил для системы нечеткого вывода Мамдани на основе числовых данных, получаемых в процессе обучения управляемой системы. Предлагаемый в статье подход к решению данной проблемы базируется на алгоритмах четкой и нечеткой кластеризации, таких как алгоритм горной кластеризации и алгоритм Густафсона–Кесселя. Он позволяет значительно упростить процесс формирования набора нечетких правил и минимизировать участие человека в этом процессе, позволяя автоматически подбирать количество правил, а также определять все необходимые параметры каждого из них. Для реализации предложенного подхода были написаны две компьютерные программы. Первая из них собирает числовые данные при управлении человеком гусеничной тележкой. На основе собранных данных эта программа строит базу нечетких правил управления гусеничным шасси. Эта база нечетких правил и ее компьютерная реализация в дальнейшем используется во второй программе для автоматизированного управления мобильным роботом на гусеничном шасси на плоскости при помощи изменения силы тяги каждой из гусениц в зависимости от положения цели, к которой робот должен приблизиться на заданное расстояние.

*Ключевые слова:* система нечеткого вывода, нечеткая кластеризация, компьютерная реализация.

## ОБРАЗЕЦ ЦИТИРОВАНИЯ

Пташко А.Е., Ухоботов В.И. Автоматическая генерация базы нечетких правил на основе числовых данных для управления мобильным роботом с гусеничным шасси // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 60–72. DOI: 10.14529/cmse170304.

## Введение

Мобильные роботы на гусеничном шасси применяются в условиях плохой проходимости. Их уравнения движения существенно отличаются от уравнений движения роботов на колесном шасси. Существуют разные методы, с помощью которых осуществляется синтез управления мобильными роботами на гусеничном шасси. Например, используются принципы и методы синергетической теории управления [1–4]. В настоящее время большое применение приобрели системы управления, для работы которых используется теория нечетких множеств [5]. Такая популярность объясняется тем, что они позволяют строить управление для задач, где по разным причинам затруднено построение точных математических моделей и алгоритмов работы с ними. Несомненно, нечеткие системы управления [6–11] по оптимальности своей работы уступают точным математическим алгоритмам, однако, зачастую построение таких алгоритмов не всегда возможно, либо их вычислительная сложность слишком велика, что делает целесообразным применение нечеткой логики.

Однако есть один очень важный момент, который затрудняет использование систем управления на основе нечеткого вывода — наборы таких правил необходимо формировать вручную. Это очень трудоемкий процесс, и порой он требует от одного или нескольких экспертов подбора огромного числа числовых параметров. Поэтому актуальной является задача минимизации участия человека, а в идеале, полной автоматизации построения базы нечетких правил и подбора необходимых параметров. Следует отметить, что существуют работы, в которых предлагаются решения такой задачи [12].

Целью работы является исследование и реализация подхода, при котором человек будет принимать участие в составлении базы правил опосредованно. Реализация подхода выполнена на примере задачи построения системы автоматизированного управления гусеничной тележкой.

Статья организована следующим образом. В первом разделе описывается задача, на примере которой был реализован предложенный подход. Во втором разделе приводится описание непосредственно самого подхода, обосновывается причина его использования. Третий раздел посвящен аспектам компьютерной реализации построения набора нечетких правил и их дальнейшего использования. В заключении приводятся основные результаты работы.

## 1. Описание задачи

Имеется гусеничная тележка (рис. 1) и некоторый объект, уравнение движения которого неизвестно (цель). Задача состоит в том, чтобы построить автоматизированное управление гусеничной тележкой на плоскости при помощи системы нечеткого вывода Мамдани [8], которое бы позволяло ему догнать цель, то есть приблизиться к ней на некоторое заданное расстояние. В данной задаче направление движения цели задается человеком, а управление тележкой осуществляется за счет изменения мощности левой и правой гусеницы. Координаты  $(x, y)$  гусеничной тележки на плоскости и угол  $\theta$  поворота его оси относительно оси  $x$  изменяются согласно следующей системе уравнений [1, 2, 4]:

$$\left\{ \begin{array}{l} \dot{x} = V \cos \theta \\ \dot{y} = V \sin \theta \\ \dot{\theta} = \omega \\ m \dot{V} = u_1 - F_{\text{тр}} \\ I_z \dot{\omega} = -\frac{B u_2}{2} - M_r \\ u_1 = f^r + f^l \\ u_2 = f^r - f^l, \end{array} \right. \quad (1)$$

где  $V$  — линейная скорость,  $\omega$  — угловая скорость,  $B$  — расстояние между гусеницами,  $m$  — масса тележки,  $I_z$  — момент инерции относительно центра тележки,  $f^r, f^l$  — силы тяги правой и левой гусеницы соответственно,  $M_r$  — момент сопротивления повороту. Сила трения  $F_{\text{тр}}$  в (1) вычисляется по формуле:

$$F_{\text{тр}} = k g m \quad (2)$$

где  $k$  — коэффициент трения поверхности, по которой движется тележка,  $g$  — ускорение свободного падения. Момент сопротивления повороту вычисляется следующим образом:

$$M_r = \frac{\mu m g l}{4} \quad (3)$$

где  $\mu$  — коэффициент сопротивления повороту тележки, который зависит от характеристик поверхности, по которой движется тележка, а также от конструкции ее гусеницы,  $l$  — длина гусеницы.

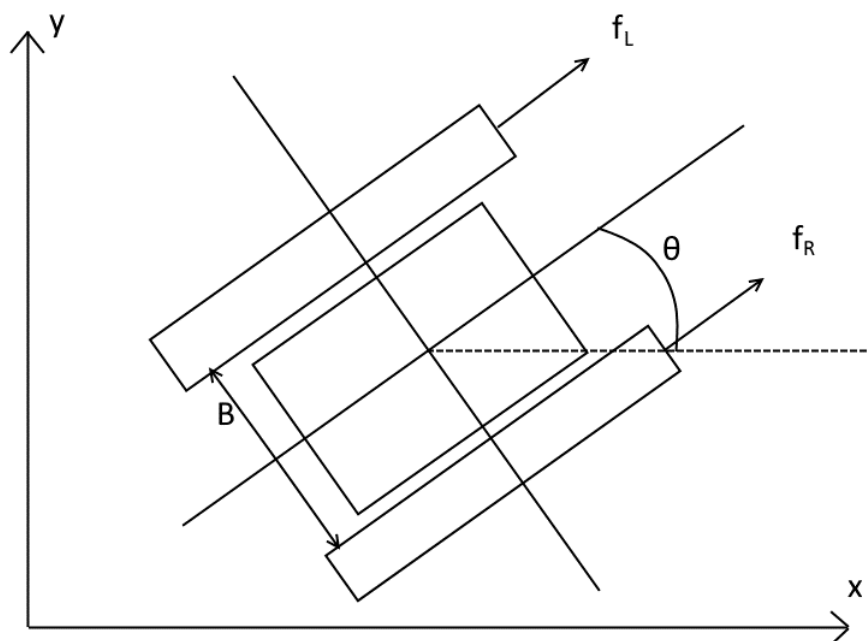


Рис. 1. Графическое представление положения гусеничной тележки на плоскости

Следует отметить, что в работе [13] приведены более полные уравнения движения, для рассматриваемых в статье управляемых систем.

## 2. Построение набора нечетких правил

Как уже было сказано выше в статье, набор нечетких правил строится на основе набора числовых данных. Получаются эти данные следующим путем: один человек управляет направлением цели, второй в это время, управляя силой тяги каждой из гусениц тележки, пытается сделать так, чтобы тележка приблизилась на заданное расстояние. В равные промежутки времени берутся координаты центра тележки, угол его поворота относительно цели и расстояние до цели. Таким образом, обучающая выборка будет выглядеть следующим образом:

$$\left\{ \left\{ f_1^r, f_1^l, a_1, d_1 \right\}, \dots, \left\{ f_n^r, f_n^l, a_n, d_n \right\} \right\}, \quad (4)$$

где  $f_n^r, f_n^l$  — силы тяги правой и левой гусеницы,  $a_i$  — угол поворота тележки относительно цели,  $d_i$  — расстояние от центра тележки до цели,  $n$  — размер обучающей выборки. Чтобы минимизировать фактор человеческой ошибки, в выборку попадают только те моменты времени, в которые расстояние до цели было меньше, либо равно расстоянию до цели в предыдущий момент времени.

Построенные в трехмерном пространстве наборы точек  $\{f_1^r, a_1, d_1\}, \dots, \{f_n^r, a_n, d_n\}$  и  $\{f_1^l, a_1, d_1\}, \dots, \{f_n^l, a_n, d_n\}$  будут выглядеть примерно так, как показано на рис. 2, а именно, будет наблюдаться концентрация обучающих примеров в некоторых областях пространства. Исходя из данного наблюдения, можно сделать вывод о том, что между

входными  $(a_i, d_i)$  и выходными  $(f_n^r, f_n^l)$  аргументами существует некоторая зависимость. Если говорить в терминах нечетких множеств, то это означает, что существует некоторое правило вывода, связывающее терм-множества аргументов, подаваемых на вход, с терм-множествами, получаемыми в результате. Это означает, что задача построения набора нечетких правил сводится к разбиению обучающей выборки на области с наиболее высокой концентрацией обучающих примеров (на кластеры).

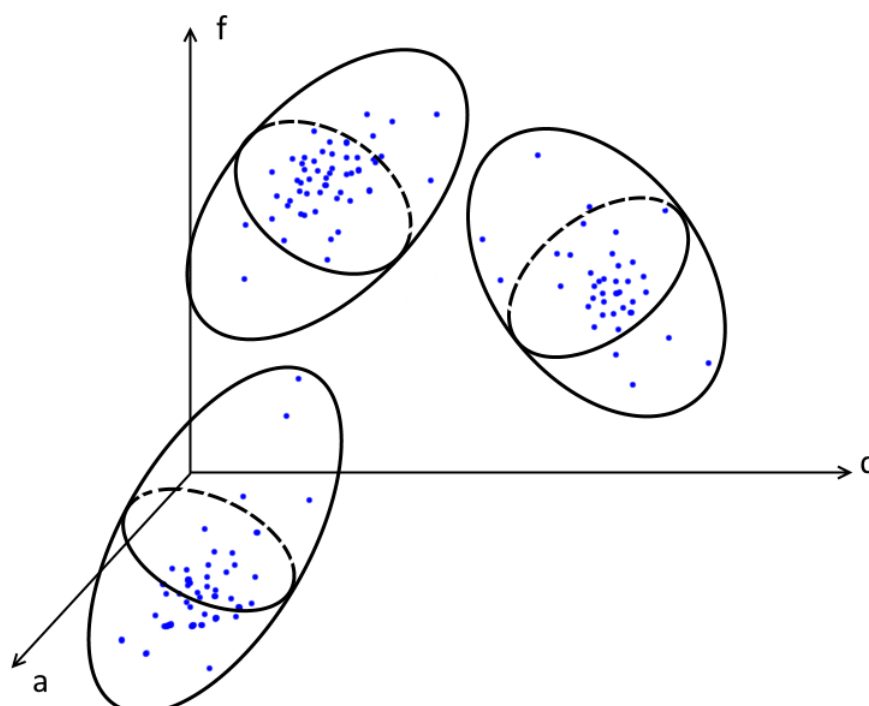


Рис. 2. Схематичное изображение обучающей выборки в пространстве

Существуют различные алгоритмы разбиения пространства на кластеры [14]. Все они делятся на алгоритмы четкой и нечеткой кластеризации. При четкой кластеризации результатом работы алгоритма является набор непересекающихся кластеров, при этом каждый элемент пространства может находиться только в одном кластере. При нечеткой кластеризации получающиеся кластеры могут пересекаться друг с другом, а каждый элемент пространства находится в каждом кластере с некоторой степенью принадлежности к нему. Очевидно, что для разбиения обучающей выборки, подходит именно нечеткая кластеризация, так как только она позволит выделить кластеры в виде нечетких множеств.

Существует несколько алгоритмов нечеткой кластеризации. Самые популярные из них — это алгоритм нечетких  $s$ -средних и алгоритм Густафсона—Кесселя [15]. Для решения задачи в статье используется второй алгоритм, так как он позволяет выделять кластеры в форме гиперэллипсов, в то время, как с помощью первого алгоритма получаются кластеры в виде гиперсфер. Кластеры в форме гиперэллипсов наиболее точно отображают связь между компонентами входа и выхода. Однако алгоритм Густафсона—Кесселя также имеет свои недостатки. Первый недостаток заключается в том, что для его работы необходимо знать количество кластеров. Второй недостаток связан с тем, что этот алгоритм очень ресурсоемкий. Для того, чтобы минимизировать проблемы с производительностью, необходимо предварительно найти приблизительные значения центров кластеров. Такие

значения ищутся в статье при помощи алгоритма горной кластеризации [16], так как он не зависит от числа кластеров и способен сам их определять.

Управление строится независимо для левой и правой гусеницы, поэтому дальнейшее описание работы алгоритмов пойдет на примере правой гусеницы. Для левой гусеницы управление строится аналогичным образом.

## 2.1. Горная кластеризация

Алгоритм горной кластеризации [16] состоит из трех шагов. На первом шаге определяются точки, которые потенциально могут быть центрами кластеров. В задаче о центрах кластеров изначально за потенциальные центры принимаются все объекты обучающей выборки. Обозначим их как  $z_i = \{f_i^r, a_i, d_i\}$ ,  $i = \overline{1, n}$ .

Второй шаг заключается в расчете так называемого потенциала каждой из точек, который показывает возможность того, что в ее окрестности может быть сформирован кластер. Точки, в окрестностях которых наиболее плотно расположены другие объекты выборки, обладают наиболее высоким потенциалом, который рассчитывается по формуле

$$P(z_i) = \sum_{k=1}^n \exp(-\alpha \cdot D(z_i, x_k)), \quad (5)$$

$$D(z_i, x_k) = \sqrt{\|z_i - x_k\|^2}.$$

В этой формуле  $P(z_i)$  — потенциал точки  $z_i$ ,  $\alpha$  — положительная константа, которая характеризует масштаб расстояния между объектами и принимается равной единице, деленной на среднее расстояние между объектами.

На третьем шаге в качестве первого центра кластера назначается точка с наибольшим потенциалом, которую назовем вершиной. В дальнейшем для оставшихся потенциальных центров кластеров потенциалы пересчитываются, без учета влияния уже найденной вершины по формуле

$$P_2(z_i) = P_1(z_i) - P_1(v_1) \exp(-\beta \cdot D(z_i, v_1)). \quad (6)$$

Здесь  $v_1$  — вершина, найденная на первом этапе алгоритма,  $P_1(z_i)$  — потенциал точки, рассчитанный на втором шаге,  $P_2(z_i)$  — новый потенциал точки.

Дальнейший процесс пересчета потенциалов и выделения центров кластеров продолжается итерационно до тех пор, пока значение находимых вершин превышает некоторый заданный порог. При решении задачи в качестве порога было выбрано значение потенциала первой вершины  $P_1(v_1)$ , деленное пополам.

Полученные при помощи алгоритма горной кластеризации центры кластеров, как было сказано выше, используются как приближительные начальные значения центров кластеров в алгоритме Густафсона—Кесселя.

## 2.2. Алгоритм Густафсона—Кесселя

Алгоритм Густафсона—Кесселя [15] является расширенным вариантом стандартного алгоритма нечетких с-средних с использованием адаптивной нормы для каждого кластера, что позволяет оптимизировать с его помощью не только координаты центров кластеров, но норм-порождающие матрицы всех кластеров. Это позволяет выделять кластеры произвольной формы.



Целью алгоритма является такое размещение центров кластеров, которое является решением следующей задачи:

$$\sum_i \sum_j u_{ij}^m d^2(x_j, c_i) \rightarrow \min, \quad 0 \leq u_{ij} \leq 1, \quad \sum_{i=1}^K u_{ij} = 1 \text{ для всех } j = \overline{1, n} \quad (7)$$

Здесь  $K$  — число кластеров,  $c_i$  — центр  $i$ -го кластера,  $m$  — степень «размытости» кластера (обычно принимается равной 2),  $u_{ij}$  — степень принадлежности  $j$ -го вектора  $i$ -му кластеру, а расстояние между вектором  $x_j$  и  $c_i$  определяется как

$$d(x_j, c_i) = \|x_j - c_i\| = \sqrt{(x_j - c_i)^T A_i (x_j - c_i)}.$$

Норм-порождающей в данном случае является симметричная положительно определенная матрица  $A_i$ , которая на начальном этапе заполняется случайными значениями из интервала  $[0, 1]$ , а начальные значения  $c_i$ , как уже упоминалось ранее, вычисляются при помощи алгоритма горной кластеризации.

С учетом применения метода неопределенных множителей Лагранжа [15] решение задачи (7) сводится к следующему алгоритму.

На первом шаге формируется матрица коэффициентов принадлежности векторов к кластерам

$$u_{ij} = \frac{1}{\sum_{k=1}^K \left( \frac{d^2(x_j, c_i)}{d^2(x_j, c_k)} \right)^{\frac{1}{m-1}}}. \quad (8)$$

Далее определяются новые координаты центров кластеров по формуле

$$c_i = \frac{\sum_{j=1}^N x_j u_{ij}^m}{\sum_{j=1}^N u_{ij}^m}. \quad (9)$$

Затем для каждого центра кластера формируется матрица ковариаций

$$s_i = \sum_{j=1}^N u_{ij}^m (x_j - c_i) (x_j - c_i)^T. \quad (10)$$

С ее помощью находятся новые масштабирующие матрицы

$$A_i = \sqrt[N]{\det(S_i)} \cdot S_i^{-1}. \quad (11)$$

Если изменения центров кластеров и матриц ковариации не превышают заранее заданного порога, то работа алгоритма считается завершенной. В противном случае их значения пересчитываются, начиная с первого шага.

### 2.3. Формирование набора нечетких правил

Разбив обучающую выборку на нечеткие кластеры и найдя центры таких кластеров, строится набор нечетких правил. Для каждого кластера все его точки проецируются на каждую ось, соответствующую входным аргументам, и значениям, получаемым на выходе (рис. 3). Для каждой такой проекции строится треугольное нечеткое число  $(l, c, r)$  [10]. Его ядром  $c$  является точка с координатой соответствующего центра кластера на данной оси. Числом  $l$  будет наименьшая координата вектора из обучающей выборки на

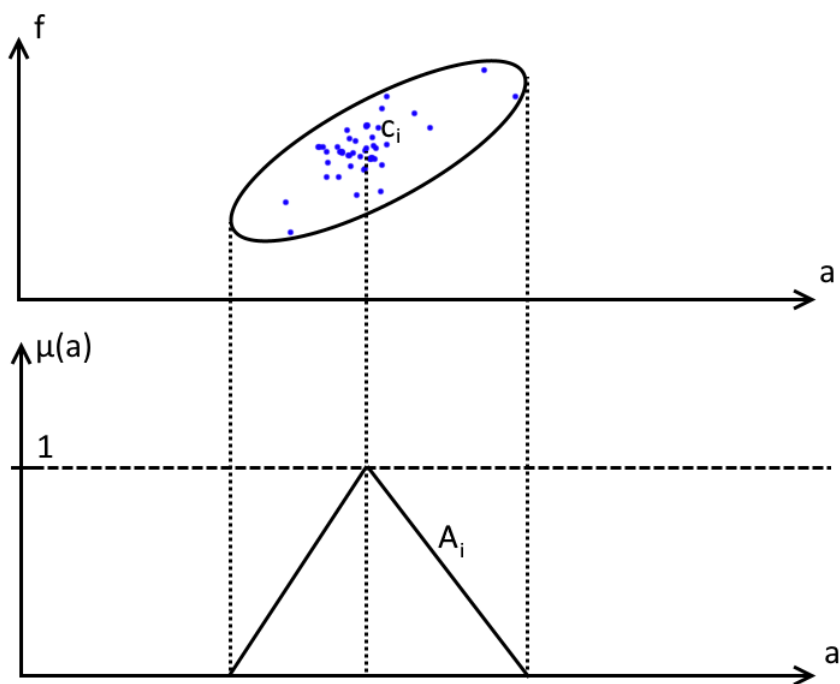


Рис. 3. Проекция кластера на ось и формирование функции принадлежности

рассматриваемой оси, попавшего в кластер. Аналогично, числом  $r$  будет наибольшая координата вектора из кластера по оси.

Построив таким образом функции принадлежности для каждого кластера, можно сформировать правила. Входным параметрам  $a_i, d_i$  и выходу  $f_i^r$  соответствуют терм-множества  $A_i, B_i, C_i$  с треугольными функциями принадлежности, найденными после проекции кластеров на соответствующие оси. Правило будет выглядеть следующим образом:

$$\text{если } a_i \text{ есть } A_i, d_i \text{ есть } B_i, \text{ то } f_i^r \text{ есть } C_i. \quad (12)$$

Возможны случаи, когда после проекции кластеров на какую-либо из осей, параметры получаемых функций принадлежности отличаются незначительно. В таком случае получаемые терм-множества объединяются. В качестве ядра берется среднее значение всех таких точек в объединяемых функциях, а в качестве чисел  $l$  и  $r$  — минимальное и максимальное значение из левых и правых границ соответственно. К примеру, терм-множество  $A_1$  с параметрами функции принадлежности  $(0; 5; 10)$  и терм-множество  $A_2$   $(0.05; 4.96; 10.1)$  будут объединены в новое множество  $A_3$   $(0; 4.98; 10.1)$ .

### 3. Программная реализация

Решение задачи управления гусеничной тележкой было реализовано в виде двух программ, написанных на языке C# в среде разработки Microsoft Visual Studio 2015. Графический интерфейс обеих программ одинаков и показан на рис. 4.

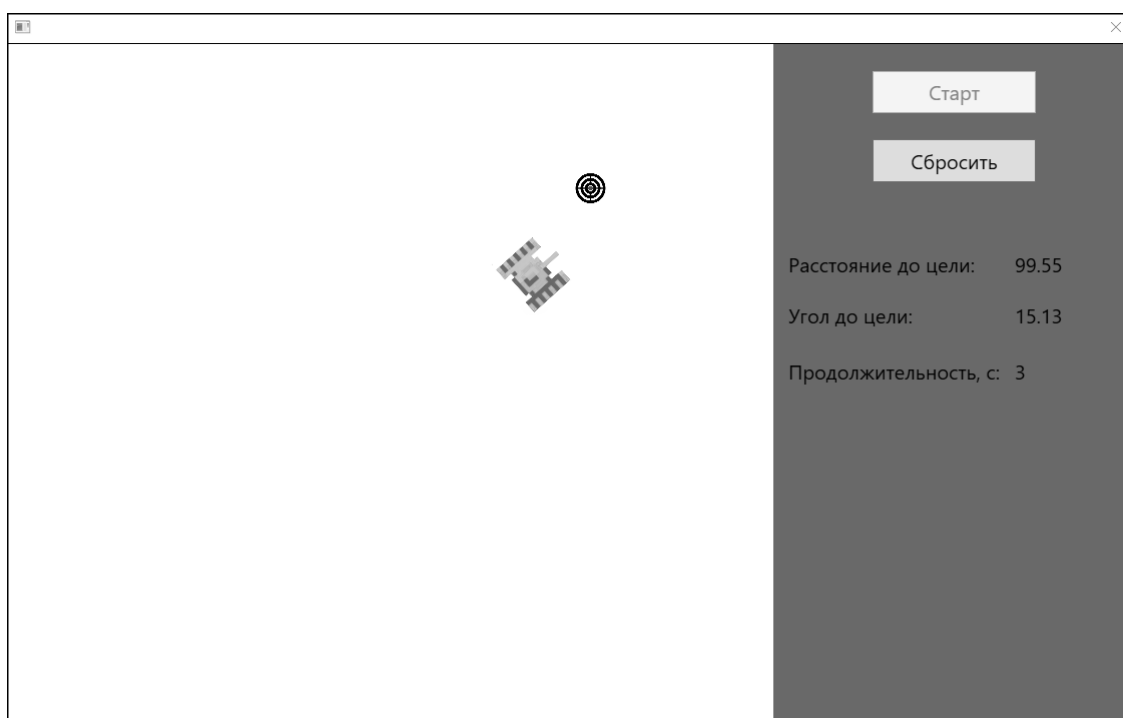


Рис. 4. Графический интерфейс реализованных программ

Первая программа предназначена для формирования обучающей выборки и формирования набора базы нечетких правил. Сбор данных осуществляется следующим образом: первый игрок при помощи компьютерной мыши задает направление цели, второй игрок в этот момент при помощи мини джойстиков на контроллере Microsoft Xbox 360 управляет силой тяги каждой из гусениц тележки таким образом, чтобы максимально приблизиться к цели. Через равные промежутки времени программа записывает в массив такие параметры, как сила тяги левой и правой гусеницы, расстояние от центра тележки до цели, угол поворота тележки относительно цели. По истечении заданного времени программа формирует на основе собранного массива данных базу нечетких правил и записывает ее в особом формате в txt файл. Пример построенных терм-множеств для нечеткой переменной angle изображен на рис. 5. Angle, distance, leftTrackPower, rightTrackPower — названия нечетких переменных (рис. 6); very\_small, big и др. — получившиеся терм-множества с параметрами соответствующих им функций принадлежности; if (angle is very\_small) and (distance is big) then (rightTrackPower is small) — получившиеся правила нечеткого вывода (рис. 7).

Вторая программа принимает на вход сгенерированный txt файл, который в дальнейшем использует для автоматизированного управления гусеничной тележкой. После нажатия на кнопку "Старт" целью все так же управляет человек, задавая направление движения посредством компьютерной мыши, а силой тяги гусениц управляет программа при помощи полученной системы нечеткого вывода.

Тестирование второй программы показало, что траектория движения тележки, управляемой при помощи сформированных нечетких правил не оптимальна, но вполне приемлема. Объясняется это тем, что правила строятся на основе данных, полученных при управлении тележки человеком, которое также далеко от идеального.

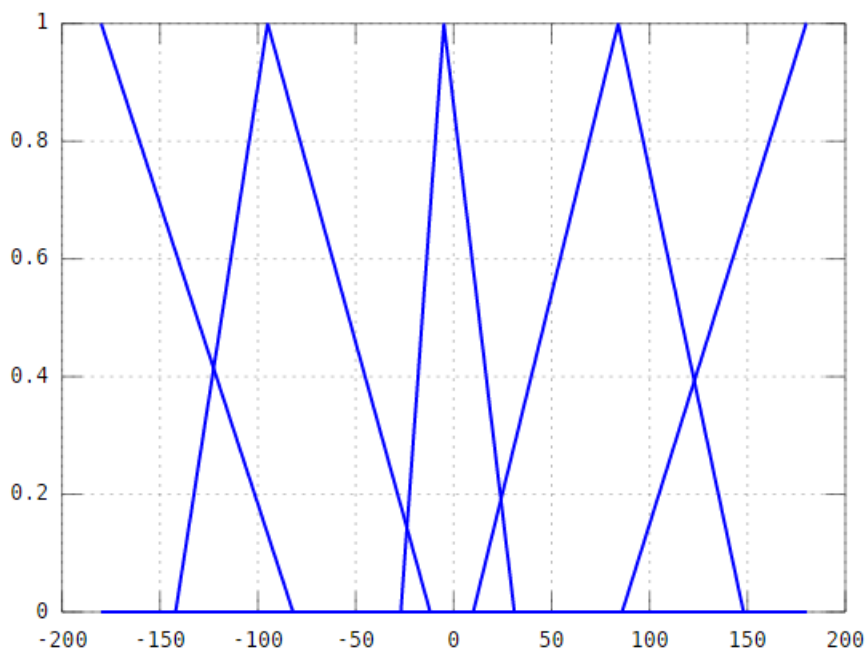


Рис. 5. Графическое изображение термов переменной angle

```

angle, -180, 180
very_small
-180, -180, -82.3621637063856
small
-142.637836293614, -95.6378362936144, -12.3621637063856
mean
-27.6378362936144, -5.63783629361439, 31.6378362936144
big
10.3621637063856, 84.3621637063856, 148.637836293614
very_big
86.3621637063856, 180, 180
    
```

Рис. 6. Пример сформированных параметров для терма angle

```

if (angle is very_small) and (distance is big) then (rightTrackPower is small)
if (angle is mean) and (distance is small) then (rightTrackPower is very_big)
if (angle is very_small) and (distance is small) then (rightTrackPower is small)
if (angle is small) and (distance is small) then (rightTrackPower is small)
if (angle is big) and (distance is small) then (rightTrackPower is very_big)
if (angle is very_big) and (distance is big) then (rightTrackPower is very_big)
if (angle is mean) and (distance is big) then (rightTrackPower is very_big)
if (angle is very_big) and (distance is small) then (rightTrackPower is very_big)
if (angle is big) and (distance is big) then (rightTrackPower is very_big)
    
```

Рис. 7. Пример сформированных нечетких правил для переменной rightTrackPower

Следует отметить, что функциональность, которая отвечает за формирование базы правил и построение управления на ее основе были реализованы в отдельной библиотеке, что означает возможность ее применения для решения схожих задач.

## Заключение

В работе исследован подход для формирования набора нечетких правил по набору числовых данных.

Работа данного подхода была продемонстрирована на примере решения автоматизированного управления гусеничной тележкой. Для этого было реализовано две программы, одна из которых позволяет человеку получить базу нечетких правил без непосредственного ручного набора этих правил и подбора необходимых параметров.

Исследованный подход позволяет минимизировать участие человека при формировании системы управления, основанной на нечетком выводе, так как позволяет автоматически определять необходимое количество нечетких правил и подбирать соответствующие им параметры. Управление, осуществляемое при помощи формируемых нечетких правил, не оптимально, однако, вполне может быть использовано, так как достаточно точно соответствует управлению со стороны человека.

В дальнейшем планируется изучение проблемы построения набора правил с функциями принадлежности разного вида (трапециевидного, сигмоидального и др.), а также разработка подходов, которые позволили бы оптимизировать параметры уже построенного набора правил.

*Работа выполнена при поддержке гранта Фонда перспективных научных исследований ФГБОУ ВО «Челябинский государственный университет» (2017 г.).*

## Литература

1. Вонг Дж. Теория наземных транспортных средств. М.: Машиностроение, 1982. 284 с.
2. Забавников Н.А. Основы теории транспортных гусеничных машин. М.: Машиностроение, 1975. 448 с.
3. Колесников А.А. Синергетическая теория управления. М.: Энергоатомиздат, 1994. 344 с.
4. Скляр А.А., Скляр С.А. Синергетическое управление мобильным роботом с гусеничным шасси // Известия ЮФУ. Технические науки. 2011. № 6. С. 118–125
5. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 161 с.
6. Castro J.L. Fuzzy Logic Controllers are Universal Approximators // IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics. Vol. 25. 1995. P. 629–635. DOI: 10.1109/21.370193
7. Kosko B. Fuzzy Systems as Universal Approximators // IEEE Transactions on Computers. 1994. Vol. 43, No. 11. P. 1329–1333. DOI: 10.1109/12.324566
8. Mamdani E.H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis // IEEE Transactions on Computing. 1977. Vol. 26. P. 1182–1191. DOI: 10.1109/TC.1977.1674779
9. Wang L.X. Fuzzy Systems are Universal Approximators // Proceedings of IEEE International Conference on Fuzzy Systems (San Diego, CA, USA, March, 8 – March, 12, 1992). IEEE Xplore Press, 1992. P. 1163–1169. DOI: 10.1109/FUZZY.1992.258721
10. Ухоботов В.И. Избранные главы теории нечетких множеств. Челябинск: Изд-во Челяб. гос. ун-та, 2011. 245 с.

11. Ухоботов В.И., Величко В.С. Моделирование заданного движения пятизвенного механизма // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2013. Т. 2. № 3. С. 104–110.
12. Cordon O., F. Herrera, P. Villar. Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base // *IEEE Transactions on fuzzy systems*, Vol. 9, No. 4, 2001. P. 667–674. DOI: 10.1109/91.940977
13. Сергеев Л.В. Теория танка. М: Военная ордена Ленина Краснознаменная академия бронетанковых войск имени маршала Советского Союза Р.Я. Малиновского, 1973. 496 с.
14. Hartigan J. Clustering Algorithms. Wiley, New York, NY, 1975. 351 p.
15. Gustafson D.E., Kessel W.C. Fuzzy Clustering With a Fuzzy Covariance Matrix // *Proceedings IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes*. (San Diego, CA, USA, January, 10 – January, 12, 1978). IEEE Xplore Press, 1978. P. 761–766. DOI: 10.1109/CDC.1978.268028
16. Chiu S., Fuzzy Model Identification Based on Cluster Estimation // *Journal of Intelligent & Fuzzy Systems*. 1994. Vol. 2, No. 3. P. 267–278. DOI: 10.3233/IFS-1994-2306

Пташко Евгений Анатольевич, аспирант, кафедра теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

Ухоботов Виктор Иванович, д.ф.-м.н., профессор, зав. каф. теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

---

DOI: 10.14529/cmse170304

## AUTOMATIC GENERATION OF FUZZY RULES FOR CONTROL OF A MOBILE ROBOT WITH TRACK CHASIS BASED ON NUMERICAL DATA

© 2017 E.A. Ptashko, V.I. Ukhobotov

*Chelyabinsk State University (st. Bratiev Kashirinikh 129, Chelyabinsk, Russia 454001)*

*E-mail: eugene.ptashko@gmail.com, ukh@csu.ru*

Поступила в редакцию: 08.06.2017

In this paper, we consider the problem of generating a set of fuzzy rules for the Mamdani fuzzy inference system based on numerical data obtained in the learning process of a managed system. The approach proposed in the article to solve this problem is based on algorithms for clear and fuzzy clustering, such as the mining clustering algorithm and the Gustafson – Kessel algorithm. It allows you to significantly simplify the process of forming a set of fuzzy rules and minimize the participation of a person in this process, allowing you to automatically select the number of rules, as well as determine all the necessary parameters for each of them. To implement the proposed approach, two computer programs were written. The first of them collects numeric data when a person manages a robot. Based on the collected data, this program builds a base of fuzzy rules for controlling mobile robot on a tracked chassis. This base of fuzzy rules and its computer implementation is further used in the second program for automated control of a mobile robot in the plane by varying the tractive force of each of the tracks depending on the position of the target to which the robot should approximate a given distance.

*Keywords: fuzzy inference system, fuzzy clustering, computer implementation.*

## FOR CITATION

Ptashko E.A., Ukhobotov V.I. Automatic Generation of Fuzzy Rules for Control of a Mobile Robot with Track Chasis Based on Numerical Data. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 60–72. (in Russian) DOI: 10.14529/cmse170304.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Wong J.Y. Theory of Ground Vehicles, 3rd Edition. Wiley-Interscience, 2001. 284 p.
2. Zabavnikov N.A. *Osnovy teorii transportnykh gusenichnykh mashin* [The Fundamentals of the Theory of Transport Caterpillar Vehicles]. M.: Mechanical Engineering, 1975. 448 p.
3. Kolesnikov A.A. *Sinergeticheskaya teoriya upravleniya* [Synergetic Theory of Management]. M.: Energoatomizdat, 1994. 344 p.
4. Scliarov A.A., Scliarov S.A. Synergetic Control of Tracked Mobile Robot. *Izvestiya YuFU. Tekhnicheskie nauki* [SFU News. Technical Science]. 2011. no. 6. pp. 118–125. (in Russian)
5. Zadeh L.A. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. American Elsevier Publishing Company, New York, 1973. 161 p.
6. Castro J.L. Fuzzy Logic Controllers are Universal Approximators. IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics. 1995. vol. 25. pp. 629–635. DOI: 10.1109/21.370193
7. Kosko B. Fuzzy Systems as Universal Approximators. IEEE Transactions on Computers. 1994. vol. 43, no. 11. pp. 1329–1333. DOI: 10.1109/12.324566
8. Mamdani E.H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. IEEE Transactions on Computing. 1977. vol. 26. pp. 1182–1191. DOI: 10.1109/TC.1977.1674779
9. Wang L.X. Fuzzy Systems are Universal Approximators. Proceedings of IEEE International Conference on Fuzzy Systems (San Diego, CA, USA, March, 8 – March, 12, 1992). IEEE Xplore Press, 1992. pp. 1163–1169. DOI: 10.1109/FUZZY.1992.258721
10. Ukhobotov V.I. *Izbrannye glavy teorii nechetkikh mnozhestv*. [Selected Chapters of the Theory of Fuzzy Sets]. Chelyabinsk, Publishing of the Chelyabinsk State University, 2011. 245 p.
11. Ukhobotov V.I. Simulation of a Given Motion of a Five-Link Mechanism. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Vychislitel'naya matematika i informatika* [Bulletin of South Ural State University. Series: Computational Mathematics and Informatics]. 2013. vol. 2, no. 3. pp. 104–110. (in Russian)
12. Cordon O., F. Herrera, P. Villar. Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base. IEEE Transactions on Fuzzy Systems. 2001. vol. 9, no. 4. pp. 667–674. DOI: 10.1109/91.940977
13. Sergeev L.V. *Teoriya tanka*. [Tank Theory]. Military Order of Lenin Red Banner Academy of Armored Forces Named After Marshal of the Soviet Union R.Ya. Malinovsky, 1973. 496 p.

14. Hartigan J. Clustering Algorithms. Wiley, New York, NY, 1975. 351 p.
15. Gustafson D.E., Kessel W.C. Fuzzy Clustering With a Fuzzy Covariance Matrix. Proceedings IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes. (San Diego, CA, USA, January, 10 – January, 12, 1978). IEEE Xplore Press, 1978. pp. 761–766. DOI: 10.1109/CDC.1978.268028
16. Chiu S., Fuzzy Model Identification Based on Cluster Estimation. Journal of Intelligent & Fuzzy Systems. 1994. vol. 2, no. 3. pp. 267–278. DOI: 10.3233/IFS-1994-2306



## ВЕКТОРНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ТЕРМОВ

© 2017 Д.В. Бондарчук

*Уральский государственный университет путей сообщения  
(620034 Екатеринбург, ул. Колмогорова, д. 66)  
E-mail: [dvbondarchuk@gmail.com](mailto:dvbondarchuk@gmail.com)  
Поступила в редакцию: 26.07.2015*

Большинство методов интеллектуального анализа текстов используют векторную модель представления знаний. Векторная модель использует частоту (вес) термина, чтобы определить его важность в документе. Термы могут быть схожи семантически, но отличаться лексикографически, что, в свою очередь, приведет к тому, что классификация, основанная на частоте термов, не даст нужного результата.

Причиной ошибок является отсутствие учета таких особенностей естественного языка, как синонимия и полисемия. Неучет этих особенностей, а именно синонимии и полисемии, увеличивает размерность семантического пространства, от которой зависит быстрдействие конечного программного продукта, разработанного на основе алгоритма. Кроме того, результаты работы многих алгоритмов сложно воспринимаются экспертом предметной области, который подготавливает обучающую выборку, что, в свою очередь, также сказывается на качестве выдачи алгоритма.

В работе предлагается модель, которая помимо веса термина в документе, так же использует «семантический вес термина». «Семантический вес термов» тем выше, чем они семантически ближе друг к другу.

Для вычисления семантической близости термов будем использовать адаптацию расширенного алгоритма Леска. Метод расчета семантической близости состоит в том, что для каждого значения рассматриваемого слова подсчитывается число слов упомянутых как в словарном определении данного значения (предполагается, что словарное определение содержит описание нескольких значений слова), так и в ближайшем контексте рассматриваемого слова. В качестве наиболее вероятного значения слова выбирается то, для которого такое пересечение оказалось больше. Векторная модель с учетом семантической близости термов решает проблему неоднозначности синонимов.

*Ключевые слова: интеллектуальный анализ текстов, векторная модель, семантическая близость .*

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Бондарчук Д.В. Векторная модель представления знаний на основе семантической близости термов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 73–83. DOI: 10.14529/cmse170305.

### Введение

Большинство методов подбора персональных рекомендаций используют векторную модель представления знаний. Задача подбора персональных рекомендаций заключается в следующем. Пусть имеется выборка текстовых данных (например, товары, услуги), формируемых пользователями, которые необходимо обработать и систематизировать. Пусть так же имеется выборка данных пользователей (например, покупателей, поставщиков) так же представленная в текстовом виде. Технических отличий между двумя этими выборками нет, различия скорее идеологические. Первая выборка — это то, что мы используем для построения модели, а вторая — то, для чего мы анализируем с использованием модели. Необходимо обработать вышеуказанные данные таким образом, чтобы можно было их использовать для быстрого подбора персональных рекомендаций для

любого пользователя. При этом данные между категориями распределены неравномерно, и пользователи независимо от их предпочтений всегда гарантированно должны получить выборку рекомендаций определенного объема. На практике очень часто встречаются текстовые корпуса, неравномерно распределенные между категориями. Категории в данном случае — это некие группы, число групп конечно и известно заранее, которые формируются специалистом в предметной области.

Векторная модель использует вес (частоту) термина, чтобы определить его важность в документе. Термы могут быть схожи семантически, но отличаться лексикографически, что в свою очередь приведет к тому, что классификация основанная на частоте термов не даст нужного результата.

*Документ* в векторной модели рассматривается как неупорядоченное множество термов. Различными способами можно определить вес термина в документе — «важность» слова для идентификации данного текста. Например, можно просто подсчитать количество употреблений термина в документе, так называемую частоту термина, — чем чаще слово встречается в документе, тем больший у него будет вес. Если терм не встречается в документе, то его вес в этом документе равен нулю.

Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если теперь для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов.

Более формально это утверждение можно представить в виде формулы:

$$\vec{d}_i = (w_{i1}, w_{i2}, \dots, w_{in}) \quad (1)$$

где  $\vec{d}_i$  — векторное представление  $i$ -го документа,  $w_{ij}$  — вес  $j$ -го термина в  $i$ -м документе,  $n$  — общее количество различных термов во всех документах коллекции.

Многие исследования [1–3] на тему решения проблемы лексической многозначности предлагают использовать семантическую связанность и семантические меры. Применяя базовые принципы семантического анализа можно улучшить производительность методов интеллектуального подбора персональных рекомендаций. Например, в реализации известной семантической сети WordNet [14], разработанной в Принстонском университете, используются так называемые «синсеты» — синонимические ряды, объединяющие слова со схожим значением. Каждый «синсет» содержит список синонимов или синонимичных словосочетаний и указатели, описывающие отношения между ним и другими «синсетами». Слова, имеющие несколько значений, включаются в несколько «синсетов» и могут быть причислены к различным синтаксическим и лексическим классам. Такой подход дает более точные результаты в сравнении с классической векторной моделью представления знаний.

Похожая техника представляет собой отображение термов документа в их «смысл» и составление функциональных векторов документа. В терминах СУБД это означает, что всем словам, имеющим один и тот же смысл приписывается некий идентификатор, который в свою очередь и становится термом. Конечно, качество обучения улучшается, однако опыт использования этой техники показывает, что улучшается оно незначительно [10, 15].

В статье предлагается другой подход — вычисление семантической близости термов (векторная модель семантической близости термов). Данная модель, помимо веса термина

в документе, так же использует «семантический вес термина». «Семантический вес термов» тем выше, чем они семантически ближе друг к другу.

Для вычисления семантической близости термов используется адаптация расширенного алгоритма Леска [4]. Данный метод состоит в следующем. Для каждого значения рассматриваемого слова подсчитывается число слов упомянутых как в словарном определении данного значения, так и в ближайшем контексте рассматриваемого вхождения слова. В качестве наиболее вероятного значения выбирается то, для которого такое пересечение оказалось больше. Векторная модель семантической близости термов решает проблему неоднозначности синонимов. Опыт использования данной модели показывает, что эффективность интеллектуального подбора персональных рекомендаций по сравнению с использованием стандартной векторной модели значительно повышается.

Целью данной работы является разработка и исследование алгоритма вычисления семантической близости и его применения для повышения эффективности подбора персональных рекомендаций. В разделе 1 настоящей статьи производится постановка задачи устранения лексической многозначности. В разделе 2 описываются известные подходы к устранению лексической многозначности. В разделе 3 описывается алгоритм переопределения классического веса термина в векторе документа для учета семантических связей между каждой парой термов. В разделе 4 описывается способ вычисления семантической близости с помощью адаптации метода Леска, основанный на использовании семантической БД WordNet. В разделе 5 приводится пример использования алгоритма на текстах различных предметных областей, а так же оценивается его эффективность. В заключении подводятся итоги исследования, описываются недостатки предложенного алгоритма и формулируются направления для дальнейших исследований.

## 1. Постановка задачи устранения лексической многозначности

Проблема снятия лексической многозначности может быть переформулирована так же, как задача максимизации с использованием формализма скрытых Марковских моделей [5]. Пусть  $T$  — множество терминов,  $M$  — множество значений, соответствующих терминам. Для последовательности терминов  $\tau = \{t_1, \dots, t_n\}$ , где  $\forall i t_i \in T$ , задача состоит в нахождении наиболее вероятной последовательности значений  $\mu = \{m_i, \dots, m_n\}$ , где  $\forall i m_i \in M$ , соответствующей входным терминам.

$$\hat{\mu} = \arg_{\mu} P(\mu|\tau) = \arg_{\mu} \left( \frac{P(\mu)P(\tau|\mu)}{P(\tau)} \right) \quad (2)$$

Поскольку вероятность  $P(\tau)$  для входной последовательности является величиной постоянной, то задача сводится к максимизации числителя, указанного в формуле (2). Для решения этого уравнения делается марковское предположение, что значение  $i$ -го термина зависит только от конечного числа значений дущих терминов [6]:

$$\hat{\mu} = \arg_{\mu} \left( \prod_{i=1}^n P(m_i|m_{i-1}, \dots, m_{i-k})P(t_i|m_i) \right) \quad (3)$$

где  $k$  — порядок модели.

Множители равенства (3) определяют скрытую Марковскую модель  $k$ -го порядка, где наблюдения соответствуют входным терминам, состояния соответствуют значениям

терминов,  $P(m_i|m_{i-1}, \dots, m_{i-k})$  — вероятность перехода между состояниями,  $P(t_i|m_i)$  — вероятность появления термина  $t_i$  в каждом состоянии  $m_i$ .

Дальнейшее использование данной модели связано со значительными трудностями, в частности с разреженностью языка. Например, чтобы построить модель перехода для Марковской модели первого порядка, необходимо оценить вероятность каждой пары состояний, что для данной задачи сводится к вероятности встречи двух терминов в конкретных значениях вместе. Для задачи устранения лексической многозначности проблема оценки параметров марковской модели является нетривиальной задачей. Это связано с большими объемами обрабатываемой информации, то есть с объемами представленных знаний и с тем, что слова в тексте на естественном языке распределяются не равномерно, а по закону Ципфа [7]

Закон Ципфа — эмпирическая закономерность распределения частоты слов естественного языка: если все слова языка упорядочить по убыванию частоты их использования, то частота  $i$ -го слова в таком списке окажется приблизительно обратно пропорциональной его порядковому номеру  $i$ .

## 2. Подходы к устранению лексической многозначности

Большинство подходов к устранению лексической многозначности связаны с развитием огромных баз знаний, созданных вручную, таких как WordNet. Можно указать на очевидный недостаток такого подхода — ограниченность области применения данных методов, поскольку ручное поддержание баз знаний в актуальном состоянии является очень сложной и трудозатратной задачей.

Если в семантической сети, на основе которой производится анализ (например, WordNET) используются различные значения для многозначных слов, то для анализа необходимо обеспечение автоматического выбора между этими многозначными сущностями [7]. Обычно в таких случаях используется наивный метод, который выбирает наиболее часто встречающуюся сущность. Очевидно, что использование такого подхода далеко от идеала и в ряде случаев может давать необъективные результаты. Частично эта проблема решается в концепции «универсального терминологического пространства», однако пока о какой-либо реализации этого пространства неизвестно.

Другим подходом к решению проблемы лексической многозначности является использование внешних источников данных. С развитием сети Интернет появилось огромное количество документов, которые связаны между собой гиперссылками. Например, в работе [8] рассматривалась возможность использования глобальной энциклопедии «Википедия» в качестве аннотированного корпуса для обучения Марковской модели. Для методов, использующих данный подход, очень часто применяют алгоритм Леска. Алгоритм основан на предположении, что многозначное слово и его окружение относятся к одной теме [4].

Все вышеописанные методы и алгоритмы так или иначе основаны на внешних данных и имеют один общий недостаток. В основе всех этих алгоритмов явно или неявно лежит предположение, что существуют однозначные термины, на основании которых в последствии определяются многозначные термины. Это в свою очередь составляет огромную проблему, поскольку в неспециализированных текстах, таких, как объявления о поиске работы, новостных статьях, участвуют только многозначные термины, либо присутствующие однозначные термины слабо связаны с темой документа и могут быть

расценены классификатором, как стоп-слова. Этот факт приводит к тому, что точность и эффективность указанных методов в значительной степени ухудшается при их применении для классификации вакансий или новостей [10].

Для избавления от лексической многозначности так же используются меры семантической связности. Отметим, что семантическая близость и семантическая связность — это разные понятия. *Семантическая близость* является частным случаем семантической связности. *Семантическая связность* — это количество связей, с помощью которых связаны два слова. Перечислим наиболее известные способы вычисления семантической связности.

1. *Мера Ликока—Чодороу*

Мера Ликока и Чодороу [12] основана на вычислении длины пути между терминами. Кратчайшим путем от одного термина к другому считается путь, который использует наименьшее количество соседних термов. Данную меру можно представить в виде следующей формулы:

$$related_{lch} = (t_1, t_2) = \max [-\log (L(t_1, t_2)/(2 \cdot D))] \quad (4)$$

где  $L(t_1, t_2)$  — кратчайшая длина пути (наименьшее количество узлов) между двумя терминами,  $D$  — максимальная глубина (максимальное количество узлов от корневого узла, либо количество узлов до ближайшего общего предка [12]).

2. *Мера Цеша*

Вычисление меры Цеша [12] подобно вычислению меры Ликока—Чодороу, основное отличие заключается в том, что поиск кратчайшего пути осуществляется между терминами по произвольным типам ссылок.

3. *Мера Лина*

Вычисление семантической связности с помощью меры Лина основано на теореме близости. Она гласит, что близость между двумя терминами можно вычислить с помощью коэффициента отношения количества текстов (корпусов), в которых термы встречаются вместе к частоте их встречаемости в определениях.

Отсутствие общих терминов между двумя документами еще не означает, что они являются абсолютно несхожими. Термины могут быть синтаксически различны, но в то же самое время семантически очень близки. Дальнейшее развитие метода анализа данных будет основано именно на этом утверждении.

### 3. Предлагаемый алгоритм

Чтобы учесть семантическую связь между терминами, вес термина в документе будем рассчитывать несколько иначе, чем в классической векторной модели представления знаний. Термином (термом) будем называть слово, обработанное с помощью стеммера Портера [9] и не содержащееся в списке стоп-слов.

Настройка весов термов производится с помощью вычисления семантической близости связанных термов. Считается, что термины связаны, если они находятся в одном документе в непосредственной близости друг к другу. Новый вес термина рассчитывается следующим образом:

$$\tilde{w}_{dt_1} = w_{dt_1} + \sum_{t_1 \neq t_2} similarity(t_1, t_2) \quad (5)$$

где  $w_{dt_1}$  — вес термина в документе  $d$  до настройки,  $similarity$  — семантическая близость терминов  $t_1$  и  $t_2$ , рассчитываемая с помощью адаптации расширенного метода Леска. Суммирование происходит по всем терминам документа  $d$ .

Этот шаг переопределяет классический вес термина в векторе документа и учитывает семантические связи между каждой парой терминов. Для вычисления исходных весов терминов в документе будем использовать меру  $tf.idf$ . Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции [11]. Мера  $tf.idf$  термина  $t$  в документе  $d$  вычисляется следующим образом:

$$tf.idf(d, t) = \ln(tf(d, t) + 1) \ln \frac{|D|}{df(t)} \quad (6)$$

где  $df(t)$  — документная частота термина, показывающая количество документов, в которых встречается терм,  $tf(t, d)$  — число появлений термина  $t$  в документе  $d$ , нормализованное общим количеством терминов в документе  $d$ ,  $|D|$  — общее количество документов.

Предлагается использовать именно эту меру, поскольку она приписывает большие веса терминам, которые редко встречаются в обучающей выборке, но часто в некоторых конкретных документах.  $tf.idf$  дает примерно на 14% более точный результат, чем стандартная мера  $tf$ , основанная на частоте термина в документе [11].

В [2] показано, что каждая из категорий обычно представлена множеством «базовых» слов, а остальные слова являются слишком общими, чтобы определять категорию. Предлагается использовать общие слова для повышения значимости (весов) «базовых» слов. Данное решение в значительной степени улучшило результаты подбора персональных рекомендаций, поскольку при использовании этого подхода определение принадлежности документа некоторой категории происходит более точно.

#### 4. Расширенный алгоритм Леска

Оригинальный алгоритм Леска [13] предусматривает использование только словарных значений анализируемого слова и же его ближайшего контекста. Это является существенным ограничением, поскольку словарные определения как правило являются очень короткими, и влияют на рассчитанную по алгоритму Леска близость слов только косвенно [13]. Если взять для примера одну из самых крупных баз знаний WordNet, то средняя длина определения слова в словаре равна всего семи словам [5].

Расширенный метод Леска расширяет определения сравниваемых слов и включает определения слов, которые связаны со сравниваемыми словами. Будем считать, что два термина похожи, если их определения содержат похожие слова. В самом простом случае расширенный метод Леска можно выразить следующей формулой:

$$\begin{aligned} similarity_{extLesk}(t_1, t_2) = & overlap(gloss(t_1), gloss(t_2)) + \\ & overlap(gloss(hyppo(t_1)), gloss(hyppo(t_2))) + \\ & overlap(gloss(hyppo(t_1)), gloss(t_2)) + \\ & overlap(gloss(t_1), gloss(hyppo(t_2))) \end{aligned} \quad (7)$$

где  $overlap(t_1, t_2)$  — количество совпадений между терминами  $t_1$  и  $t_2$ ,  $gloss(t)$  — определение термина  $t$ ,  $hyppo(t)$  — гипероним слова, например для слова «красный» гиперонимом является слово «цвет»,  $t_1$  и  $t_2$  — термины.

В классической версии алгоритма Леска гиперонимы не используются, однако их использование значительно улучшает качество выдачи алгоритма. В работах [3] и [6] используются синсеты и гиперонимы из английской версии WordNET, но синсет выбирается согласно наивному методу, после чего выбирается соответствующий синсету гипероним. Эксперименты проводились на нескольких независимых общедоступных выборках. Авторы данных работ сделали вывод, что использование гиперонимов привело к улучшению качества работы классификатора на всех обучающих множествах. Кроме того, выяснилось, что применение гиперонимов почти всегда улучшает качество работы классификатора по сравнению с применением только синсетов [3].

Гиперонимы слов для первоначальной оценки можно взять, например, в российской версии WordNet, разработка которой осуществляется в Петербургском университете путей сообщения.

## 5. Оценка результатов

В исходный алгоритм [15] внесены изменения, позволяющие повысить эффективность метода и качество результирующей выборки. Данные изменения характеризуются иным способом вычисления весов термов на этапе построения векторных моделей.

Для оценки эффективности алгоритма классификации представления знаний, использующую семантическую близость термов, по сравнению с исходным алгоритмом, была проверена работа алгоритмов на различных множествах текстов: объявления о работе, новости, литературные аннотации.

В табл. 1 представлены сведения о выборках.

Таблица 1

Сведения о выборках

Выборка	Кол-во текстов	Кол-во кат.	Распределение
Объявления о работе	около 700 тыс.	17	неравномерное
Новости	около 1,2 млн.	10	равномерное
Литературные аннотации	около 20 тыс.	13	равномерное

В качестве мер оценок результатов использовались *F-measure* и *purity*:

$$F\text{-measure} = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}, \quad (8)$$

где *precision* — количество правильных результатов в выдаче алгоритма, *recall* — общее количество результатов выдачи.

$$\textit{purity}(W, C) = \sum_k \max_j |w_k \cup c_j|, \quad (9)$$

где  $W$  — множество документов,  $w_k$  —  $k$ -ый документ,  $C$  — множество категорий (множество документов, отнесенных классификатором к категории  $k$ ),  $c_j$  — множество документов, отнесенных к категории  $j$  экспертом.

В табл. 2 представлены результаты данных оценок. Взяты средние значения оценок 15 текстов.

Таблица 2

Оценка результатов работы алгоритма классифиции

	Исходный алгоритм		Предлагаемый алгоритм	
	F-measure	Purity	F-measure	Purity
Объявления о работе	0,31	0,33	0,65	0,66
Новости	0,56	0,58	0,61	0,64
Литературные аннотации	0,56	0,57	0,63	0,67

Результаты экспериментов показывают, что использование векторной модели с вычислением семантической близости помогает улучшить результаты работы классификатора по сравнению с векторной моделью без учета семантической близости. Меры, описанные формулами (8) и (9), показывают, насколько результаты работы предложенного классификатора соответствуют действительности. Чем выше значение оценки, тем качественнее работа алгоритма.

В среднем категориальная векторная модель с использованием семантической близости дает на 8–10% более точный результат. Это связано с тем, что исходная модель менее чувствительна к «шумам» за счет настройки весовых коэффициентов с помощью вычисления семантической близости. Новые весовые коэффициенты векторов документов учитывают контекст появления термов. Более высокие веса связаны с термами, которые сильнее семантически связаны с другими термами.

Эксперименты были проведены над выборками разного рода и объема, на всех из них метод отработал эффективно. Также часть выборок была распределена неравномерно, метод и на них показал хороший результат, в то время как результаты векторной модели без учета семантической близости термов оказались неудовлетворительными.

## Заключение

В статье была рассмотрена задача формирования персональных рекомендаций и предложена векторная модель представления знаний, использующая семантическую близость термов, использование которой улучшает качество работы автоматического классификатора. Модель помогает решить проблему лексической неоднозначности терминов, а так же находит скрытые семантические связи между документами, сравнивая семантически близкие термины.

Проведенные эксперименты показали, что метод, основанный на использовании этой модели, показал достаточно высокую эффективность. Среди направлений дальнейших исследований можно выделить исследование возможности статистического определения семантической близости между термами, поиск альтернативного алгоритма стемминга, который лучше работает с русским языком. Кроме того, недостаточно изученным является вопрос возможности использования в алгоритмах формирования персональных рекомендаций существующих словарей, тезаурусов и баз данных интернета.

## Литература

1. Budanitsky A., Hirst G. Evaluating WordNet-based Measures of Lexical Semantic Relatedness // Computational Linguistics. 2006. Vol. 32. P. 13–47.



2. Hotho A., Staab S., Stumme G. WordNet Improve Text Document Clustering // SIGIR 2003 Semantic Web Workshop (Toronto, Canada, July 28 – August 1, 2003). P. 541–544. DOI: 10.1145/959258.959263.
3. Sedding J., Dimitar K. WordNet-based Text Document Clustering // COLING 2004, 3rd Workshop on Robust Methods in Analysis of Natural Language Data (Geneva, Switzerland, August 23 – 27, 2004). P. 104–113. DOI: 10.3115/1220355.1220356.
4. Lesk M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone // SIGDOC'86. Proceedings of the 5th Annual International Conference on Systems Documentation (Toronto, Canada, June 8 – 11, 1986). P. 24–26. DOI: 10.1145/318723.318728.
5. Loupy C., El-Beze M., Marteau P.F. Word Sense Disambiguation Using HMM Tagger // Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC (Toronto, Canada, June 8 – 11, 1998). P. 1255–1258. DOI: 10.3115/974235.974260.
6. Jeh G., Widom J. SimRank: a Measure of Structural-Context Similarity // Proceedings of the 8th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining (Edmonton, Canada, July 23 – 25, 2002). P. 271–279. DOI: 10.1145/775047.775049.
7. Kechedzhy K.E., Usatenko O., Yampolskii V.A. Rank Distributions of Words in Additive Many-step Markov Chains and the Zipf law // Physical Reviews E: Statistical, Nonlinear, Biological, and Soft Matter Physics. 2005. Vol. 72. P. 381–386.
8. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation // Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (New York, USA, April 22 – 27, 2007). P. 196–203. DOI: 10.3115/1599081.1599184.
9. Willett P. The Porter Stemming Algorithm: Then and Now // Program: Electronic Library and Information Systems. 2006. Vol. 4, No. 3. P. 219–223.
10. Бондарчук Д.В. Выбор оптимального метода интеллектуального анализа данных для подбора вакансий // информационные технологии моделирования и управления. 2013. № 6(84). С. 504–513.
11. Salton G. Improving Retrieval Performance by Relevance Feedback // Readings in Information Retrieval. 1997. Vol. 24. P. 1–5.
12. Tan P. N., Steinbach M., Kumar V. Top 10 Algorithms in Data Mining // Knowledge and Information Systems. 2008. Vol. 14, No. 1. P. 1–37. DOI: 10.1007/s10115-007-0114-2.
13. Banerjee S., Pedersen T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet // Lecture Notes In Computer Science (Canberra, Australia, February 11 – 22, 2002). Vol. 2276. P. 136–145. DOI: 10.1007/3-540-46035-7 22.
14. Тезаурус WordNET. URL: <https://wordnet.princeton.edu/> (дата обращения: 05.02.2017).
15. Бондарчук Д.В. Интеллектуальный метод подбора персональных рекомендаций, гарантирующий получение непустого результата // Информационные технологии моделирования и управления. 2015. № 2(92). С. 130–138.

Бондарчук Дмитрий Вадимович, к.ф.-м.н., доц., Уральский государственный университет путей сообщения (Челябинск, Российская Федерация)

## VECTOR SPACE MODEL OF KNOWLEDGE REPRESENTATION BASED ON SEMANTIC RELATEDNESS

© 2017 D.V. Bondarchuk

*Ural State University of Railway Transport  
(st. Kolmogorova 66, Yekaterinburg, 620034 Russia)*

*E-mail: [dvbondarchuk@gmail.com](mailto:dvbondarchuk@gmail.com)*

Received: 26.07.2015

Most of text mining algorithms uses vector space model of knowledge representation. Vector space model uses the frequency (weight) of term to determine its importance in the document. Terms can be semantically similar but different lexicographically, which in turn will lead to the fact that the classification is based on the frequency of the terms does not give the desired result.

Analysis of a low-quality results shows that errors occur due to the characteristics of natural language, which were not taken into account. Neglect of these features, namely, synonymy and polysemy, increases the dimension of semantic space, which determines the performance of the final software product developed based on the algorithm. Furthermore, the results of many complex algorithms perceived domain expert to prepare training sample, which in turn also affects quality issue algorithm.

We propose a model that in addition to the weight of a term in a document also uses semantic weight of the term. Semantic weight terms, the higher they are semantically closer to each other.

To calculate the semantic similarity of terms we propose to use a adaptation of the extended Lesk algorithm. The method of calculating semantic similarity lies in the fact that for each value of the word in question is counted as the number of words referred to the dictionary definition of this value (assuming that the dictionary definition describes several meanings of the word), and in the immediate context of the word in question. As the most probable meaning of the word is selected such that this intersection was more. Vector model based on semantic proximity of terms solves the problem of the ambiguity of synonyms.

*Keywords: text-mining, vector space model, semantic relatedness .*

### FOR CITATION

Bondarchuk D.V. Vector Space Model of Knowledge Representation Based on Semantic Relatedness. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 73–83. (in Russian) DOI: 10.14529/cmse170305.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

### References

1. Budanitsky A., Hirst G. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*. 2006. vol. 32. pp. 13–47.
2. Hotho A., Staab S., Stumme G. WordNet Improve Text Document Clustering. *SIGIR 2003 Semantic Web Workshop (Toronto, Canada, July 28 – August 1, 2003)*. pp. 541–544. DOI: 10.1145/959258.959263.
3. Sedding J., Dimitar K. WordNet-based Text Document Clustering. *COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data (Geneva, Switzerland, August 23 – 27, 2004)*. pp. 104–113. DOI: 10.3115/1220355.1220356.
4. Lesk M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to

- Tell a Pine Cone from an Ice Cream Cone. *SIGDOC '86: Proceedings of the 5th Annual International Conference on Systems Documentation (Toronto, Canada, June 8 – 11, 1986)*. pp. 24–26. DOI: 10.1145/318723.318728.
5. Loupy C., El-Beze M., Marteau P.F. Word Sense Disambiguation Using HMM Tagger. *Proceedings of the 1st International Conference on Language Resources and Evaluation (Toronto, Canada, June 8 – 11, 1998)*. pp. 1255–1258. DOI: 10.3115/974235.974260.
  6. Jeh G., Widom J. SimRank: a Measure of Structural-context Similarity. *Proceedings of the 8th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining (Edmonton, Canada, July 23 – 25, 2002)*. pp. 271–279. DOI: 10.1145/775047.775049.
  7. Kechedzhy K.E., Usatenko O., Yampolskii V.A. Rank Distributions of Words in Additive Many-step Markov Chains and the Zipf law. *Physical Reviews E: Statistical, Nonlinear, Biological, and Soft Matter Physics*. 2005. vol. 72. pp. 381–386.
  8. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation. *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (New York, USA, April; 22 – 27, 2007)*. pp. 196–203.
  9. Willett P. The Porter Stemming Algorithm: Then and Now. *Program: Electronic Library and Information Systems*. 2006. Vol. 4., No. 3. P. 219–223.
  10. Bondarchuk D.V. Choosing the Best Method of Data Mining for the Selection of Vacancies. *Informacionnye Tehnologii Modelirovaniya i Upravleniya [Information Technology Modeling and Management]*. 2013. no. 6(84). pp. 504–513. (in Russian)
  11. Salton G. Improving Retrieval Performance by Relevance Feedback. *Readings in Information Retrieval*. 1997. Vol. 24. pp. 1–5.
  12. Tan P. N., Steinbach M., Kumar V. Top 10 Algorithms in Data Mining. *Knowledge and Information Systems*. 2008. vol. 14. no. 1. pp. 1–37. DOI: 10.1007/s10115-007-0114-2.
  13. Banerjee S., Pedersen T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *Lecture Notes In Computer Science*. 2002. vol. 2276. pp. 136–145.
  14. Tezaurus WordNET [Thesaurus WordNET]. Available at: <https://wordnet.princeton.edu/> (accessed: 05.02.2017).
  15. Bondarchuk D.V. Intelligent Method of Selection of Personal Recommendations, Guarantees a Non-empty Result. *Informacionnye Tehnologii Modelirovaniya i Upravleniya [Information Technology Modeling and Management]*. 2015. no. 2(92). pp. 130–138. (in Russian)

## ОБНАРУЖЕНИЕ И ЛОКАЛИЗАЦИЯ СТРОЕНИЙ ЗАДАННОЙ ФОРМЫ НА АЭРОСНИМКАХ В ИНФРАКРАСНОМ ДИАПАЗОНЕ

© 2017 А.В. Дунаева<sup>1,2</sup>, Ф.А. Корнилов<sup>1</sup>

<sup>1</sup> ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН

(620990 Екатеринбург, ул. Софьи Ковалевской, д. 16),

<sup>2</sup> ФГБОУ ВО Уральский федеральный университет

им. первого Президента России Б.Н. Ельцина

(620002 Екатеринбург, ул. Мира, д. 19)

E-mail: [maryanova.av@yandex.ru](mailto:maryanova.av@yandex.ru), [kornilovfa@imm.uran.ru](mailto:kornilovfa@imm.uran.ru)

Поступила в редакцию: 24.07.2017

В работе рассматривается задача обнаружения и локализации строений заданной формы на аэроизображениях земной поверхности в инфракрасном диапазоне с использованием аппарата контурного анализа. Приводится описание модификации алгоритма обобщенного преобразования Хафа для обнаружения контуров, заданных небольшим количеством параметров. Идея предлагаемого метода заключается в построении двумерных аккумуляторных массивов для каждого набора параметров фигуры в зависимости от ее положения, и их последующего объединения в результирующий аккумуляторный массив. Заполнение массивов осуществляется на основе значений модулей градиентов яркостей исходного изображения с учетом близости рассматриваемого контура к заданной форме. Близость формы определяется путем морфологического анализа контуров, найденных с помощью алгоритма Канни. Фильтрация обнаруженных объектов на основе плотности границ в их внутренней области, а также соотношения средних яркостей внутри и снаружи контура обеспечивает высокую чувствительность к заданным типам объектов и уменьшает количество ложных срабатываний алгоритма. Качество работы метода проверено на задаче локализации малоэтажных построек прямоугольной формы. Полученные результаты позволяют судить о применимости предложенного подхода для решения практических задач распознавания объектов местности.

*Ключевые слова:* обработка изображений, обнаружение объектов, контурный анализ, математическая морфология.

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Дунаева А.В., Корнилов Ф.А. Обнаружение и локализация строений заданной формы на аэроснимках в инфракрасном диапазоне // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 84–100. DOI: 10.14529/cmse170306.

### Введение

Визуальная информация является важнейшим источником информации об окружающем мире. Когда развитие вычислительной техники позволило создавать достаточно сложные системы, значительные усилия были направлены на решение задач обработки и анализа изображений. Последовавшее за этим усложнение как аппаратных, так и программных средств привело к тому, что объем входных данных стал чрезмерным для обработки и хранения, и встал вопрос о его уменьшении путем автоматизации процесса выделения наиболее значимых в контексте конкретной ситуации признаков, деталей, объектов и т.д. Некоторые из этих признаков носили физический характер (яркость, текстура) и могли быть верифицированы визуально. Другие представляли собой различные математические конструкции, операции обработки входного сигнала,

позволяющие выделить данные, недоступные для непосредственного наблюдения (спектры сигналов и др.) [1, 2]. Одним из способов уменьшения размерности стал контурный анализ.

Контур — это совокупность точек границы объекта, отделяющая его от фона. Будучи не связанным напрямую с яркостями пикселей изображения, контур является признаком, инвариантным как к изменениям яркости и контраста, так и к некоторым геометрическим искажениям: поворотам, сдвигам и изменениям масштаба [3, 4]. При этом предполагается, что контур содержит достаточно информации, необходимой для описания формы объекта. Ключевым фактором для эффективного применения данного предположения в различных задачах является отсутствие необходимости рассматривать внутренние точки объекта, что сразу переводит задачу из разряда двухмерных к значительно более простой одномерной постановке.

Контурный анализ является совокупностью методов выделения, описания, преобразования и распознавания контуров объектов на изображениях [5]. Под контуром с точки зрения поля значений яркости изображения понимается пространственно-протяженный разрыв (резкое изменение) значений яркости, который кодируется последовательностью, состоящей из комплексных чисел. Для этого выполняется обход контура (например, по часовой стрелке), начиная с некоторой зафиксированной начальной точки, и каждый вектор смещения относительно предыдущей точки записывается соответствующим комплексным числом. Такое кодирование, очевидно, инвариантно к сдвигам. Более того, имея два подобных комплекснозначных вектора, соответствующих двум контурам, можно вычислить для них нормированное скалярное произведение, которое будет тем ближе к единице, чем более похожи по форме исходные контура. Данное утверждение будет справедливо независимо от угла поворота контуров относительно друг друга. Однако, это свойство не сохраняется при рассмотрении действительных векторов координат пикселей контуров, что обуславливает необходимость перехода к комплекснозначному представлению.

Использование нормированного скалярного произведения позволяет эффективно сравнивать между собой контура (например, входной с контуром из базы данных). Однако его применение будет затруднено из-за проблематичности выделения контура на изображении. Во-первых, контур может быть виден лишь частично по причине перекрытия объектами сцены друг друга. Во-вторых, реальные изображения, с одной стороны, неизбежно содержат шум, а с другой — объекты на них зачастую незначительно отличаются по цвету и текстуре от фона, из-за чего становится крайне сложно выделить контура и произвести дальнейший анализ их формы.

В данной работе представлен подход, позволяющий выделить на изображении контура заданной формы, соответствующие определенным типам объектов. В качестве исходной задачи было выбрано дешифрирование видовой аэрокосмической съемки в инфракрасном диапазоне; в частности, рассматривается задача обнаружения строений прямоугольной формы. Изображения земной поверхности характеризуются изменчивостью отображения объектов сцены, которая, к тому же, отличается высокой структурной сложностью. В то же время, изображения инфракрасного диапазона обладают как слабой текстурированностью — что, в целом, является плюсом, поскольку значительно уменьшает объем не имеющей ценности для рассматриваемой задачи информации, так и слабой контрастностью — это, в свою очередь, затрудняет выделение перепадов яркости.

Статья организована следующим образом. В разделе 1 описаны стандартные подходы и методы выделения и анализа контуров изображений: предварительная обработка, обнаружение градиентов яркостей, прореживание контуров и определение их формы. В разделе 2 предложен алгоритм обнаружения строений заданной формы, основанный на модификации преобразования Хафа [6]. Раздел 3 посвящен деталям реализации алгоритма и результатам вычислительного эксперимента на реальных данных. В заключении подводятся итоги и намечаются направления дальнейшей работы.

## 1. Методы выделения и анализа границ объектов

Выделение перепадов яркости находит широкое применение в задачах обработки изображений. Помимо указанного во введении сравнения формы объектов для целей распознавания и классификации, это может быть сегментация — выделение границ между однородными по структуре областями, обнаружение изменений в составе объектов сцены и т.д. [7]. Благодаря такому многообразию прикладных задач было разработано большое количество различных подходов для выделения контуров объектов. Здесь мы в общих чертах представим лишь некоторые базовые из них.

Алгоритм контурного анализа состоит из следующих шагов:

1. Сглаживание входного изображения.
2. Поиск, прореживание и бинаризация модулей градиентов яркости.
3. Фильтрация и анализ найденных контуров.

Для решения задач контурного анализа необходимо предварительно сгладить исходное изображение. Рассматривая границу как перепад яркости между соседними пикселями, стоит учитывать, что, во-первых, по причине сложности сцены подобных границ будет множество, а во-вторых шум камеры, оптические и геометрические искажения приведут к тому, что практически никакие два соседних пикселя не будут иметь одинаковую яркость. Сглаживание изображения помогает заметно уменьшить количество таких перепадов. Однако, применение операции сглаживания неизбежно размывает и истинные границы. Для борьбы с этим недостатком разработан ряд методов, подавляющих шум с сохранением точности границ. Одним из классических является билатеральный фильтр [8], но в данной работе был использован более качественный *управляемый фильтр* (guided filter) [9]. Данный метод отличается простотой и гибкостью реализации, а также высоким качеством работы.

Сгладив исходное изображение, можно перейти к поиску перепадов яркости. Особенности формирования изображения на матрице камеры, а также шум, приводят к тому, что граница между объектом и фоном, как правило, представляет собой не «ступеньку», а плавное изменение яркости. Для обнаружения таких градиентов можно воспользоваться оператором Собела [10], который представляет собой дискретный дифференциальный оператор, основанный на свертке фрагментов изображения со специальными фильтрами, выделяющими перепады яркости в вертикальном и горизонтальном направлении. Для размера окна  $3 \times 3$  пикселя маски фильтра выглядят следующим образом:

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * Image, \quad Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * Image,$$

где  $Image$  — исходное полутоновое 8-битное изображение, символом  $*$  обозначен оператор свертки. На основе  $Sobel_x$  и  $Sobel_y$  можно вычислить модуль вектора градиента и его направление:

$$Sobel = \sqrt{Sobel_x^2 + Sobel_y^2}, \quad \Theta = \arctg \frac{Sobel_y}{Sobel_x}. \quad (1)$$

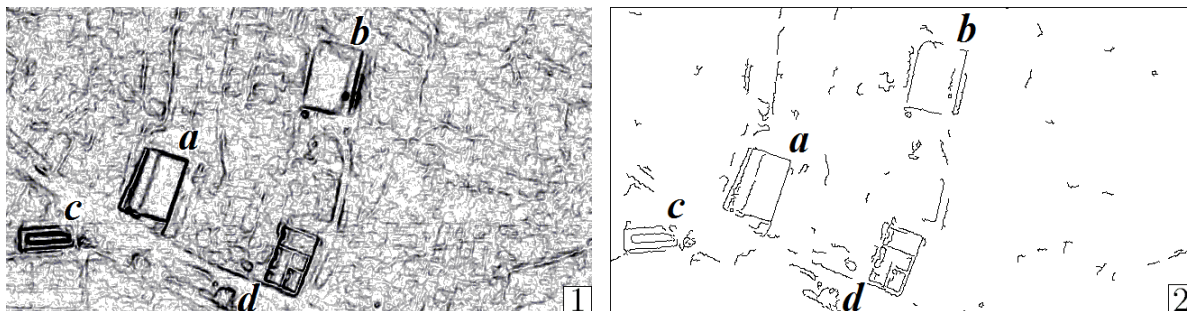


Рис. 1. Примеры работы оператора Собела (1) и алгоритма Канни (2). Для оператора Собела чем темнее точка, тем больше в ней значение модуля градиента. Найденные с помощью алгоритма Канни контура показаны черным цветом

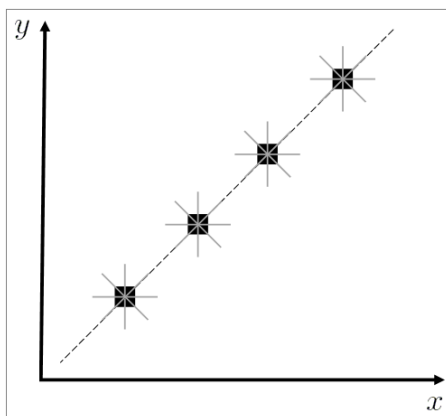
Однако оператор Собела выдает не столько контура, сколько области градиентов: иными словами, широкие полосы (толщиной в несколько пикселей) вокруг истинной границы (рис. 1(1)). Для того, чтобы получить границу как линию толщиной в один пиксель, требуется применить алгоритм Канни [11], состоящий из следующих шагов.

1. *Сглаживание*: как правило, размытие фильтром Гаусса с маской  $5 \times 5$ .
2. *Поиск градиентов*: с помощью оператора Собела, описанного выше.
3. *Подавление немаксимумов*: рассматривая известное направление градиента  $\Theta$  (1), оставляем вдоль него лишь точку, имеющую максимальное значение градиента  $Sobel$  (1). Благодаря этому шагу широкая полоса градиента, найденная с помощью оператора Собела, сжимается до линии, толщиной в один пиксель.
4. *Применение двух порогов*: задаются два пороговых значения, и точки, в которых значение градиента выше большего из них, считаются надежными границами, а те, что ниже меньшего отбрасываются. Точки, значение градиента в которых оказалось между двумя порогами, отбрасываются, если они не связаны (между ними нет пути, проходящего по точкам с ненулевым градиентом) с надежными границами.

Использование алгоритма Канни позволяет найти все существенные границы на изображении (рис. 1(2)). Однако, в конкретных приложениях, как правило, требуется обнаружить не все границы, а только принадлежащие определенным объектам, причем известно, что эти объекты характеризуются заранее известной формой. Преобразование Хафа [6] является одним из методов поиска кривых на бинарных изображениях. Параметры искомой кривой образуют пространство, в котором осуществляется поиск локальных максимумов. Заполнение пространства параметров (или аккумуляторного пространства) происходит с помощью процедуры голосования.

Первоначальный вариант преобразования Хафа разработан для поиска прямых линий. Если прямая задана уравнением  $x \cos \phi + y \sin \phi = \rho$  с параметрами  $\rho$  и  $\phi$ , то пространство параметров будет двумерным. На рис. 2 представлен пример, состоящий из четырех точек границ, выделенных на исходном изображении. Каждая точка голосует за набор прямых, которые потенциально могут пройти через нее с заданным шагом по углу (например,

четыре, если выбран шаг по углу, равный 45 градусов). Накапливая таким образом значения, получаем массив (аккумулятор), локальные максимумы которого соответствуют искомым прямым. На рис. 2 этот максимум будет соответствовать прямой, проходящей через все четыре точки. Аналогичным образом можно искать любые аналитически заданные кривые: например, окружности, для которых пространство параметров будет трехмерным.



**Рис. 2.** Поиск прямых линий с помощью преобразования Хафа. Перебирая для каждого пикселя исходного изображения возможные прямые с шагом по углу 45 градусов, получим, что каждый пиксель голосует за четыре возможных прямых. Прямая, проходящая через все четыре точки, соберет четыре голоса, в то время как все остальные прямые соберут только по одному

Для случая, когда мы ищем не прямую, а некоторую кривую, либо описывающуюся слишком большим количеством параметров, либо чье аналитическое выражение неизвестно, были разработаны различные модификации: обобщенное [3] и рекуррентное [12] преобразования Хафа. Обобщенное преобразование Хафа использует принцип сравнения с шаблоном (эталонным образцом). Для этого осуществляется поиск положения заданной шаблоном искомой кривой на изображении: выбираются центр фигуры (например, центр тяжести) и множество точек на границе, после чего строится *просмотровая таблица*, в которой для каждой граничной точки записывается расстояние от выбранного центра и угол между вектором из центра до этой точки и вектором нормали к кривой. Таким образом осуществляется запоминание формы фигуры. При таком представлении каждая точка исходного изображения отождествляется с точками просмотрной таблицы и голосует за набор возможных центров фигуры.

Можно заключить, что преобразование Хафа по своей сути является алгоритмом перебора всех возможных положений искомой кривой для всех точек контуров исходного изображения и выбора наиболее часто встречающегося набора (или наборов) параметров этой кривой. Очевидно, что, во-первых, такой перебор для достаточно сложной по форме кривой будет чрезвычайно затратным по времени и ресурсам, так как будет требовать построения пространства параметров значительной размерности. Во-вторых, сами контура входного изображения отличаются сложностью и разнообразием: с одной стороны, их очень много (например, при поиске крыши дома придется неизбежно столкнуться с контурами лифтовых шахт, вентиляционных окон и других мелких объектов, загромождающих крышу и сбивающих алгоритм). С другой стороны, сам периметр крыши из-за геометрических и яркостных искажений скорее всего не будет иметь строго прямоугольную форму. Для



преодоления указанных ограничений в следующем пункте будет представлен алгоритм, основанный на преобразовании Хафа и решающий задачу обнаружения строений заданной формы.

Помимо указанных выше, существуют другие методы выделения и анализа контуров. Следует упомянуть детекторы угловых (характерных) точек: Харриса [13], FAST [14] и других. Однако, их применение к инфракрасным изображениям затруднено вследствие значительного размытия границ объектов. Также ряд работ посвящен поиску строений на последовательностях аэроизображений в видимом диапазоне, например [15].

## 2. Алгоритм локализации строений заданной формы

Изложенная в данной работе идея применима к поиску строений любой формы, лишь необходимо, чтобы эта форма была параметризуема некоторым небольшим количеством параметров. При этом не требуется наличия аналитического выражения: в общем случае искомый контур можно задать либо последовательностью точек с привязкой к некоторому общему центру (например, центру масс), аналогично просмотрным таблицам в обобщенном преобразовании Хафа, либо как комплекснозначный вектор смещений, как было упомянуто во введении.

Введем основные обозначения. Через  $Image(x, y)$  обозначим исходное полутоновое изображение;  $Sobel(x, y)$  — изображение модулей градиентов яркости  $Image$  (1);  $Canny(x, y) \in \{0, 1\}$  — изображение, содержащее контура  $Image$ , найденные с помощью алгоритма Канны, где значение 1 соответствует точкам контуров.

Работу алгоритма будем рассматривать на примере задачи локализации строений прямоугольной формы. Обозначим через  $P = P_{w,h,\alpha}(x, y)$  — прямоугольник с центром в точке  $(x, y)$ , размером  $w \times h$  пикселей, повернутый на угол  $\alpha$ . Везде далее будем полагать, что  $w \geq h$ :

$$\begin{cases} w \in [l_{min}, l_{max}], \\ h \in [l_{min}, w], \\ \alpha \in [0, \pi], \end{cases} \quad (2)$$

с заданным шагом по каждому параметру. Параметры  $l_{min}$  и  $l_{max}$  определяются исходя из требований конкретной задачи, и при наличии априорной информации о размере искомых объектов можно значительно сократить перебор, тем самым уменьшив как время работы, так и количество ложных срабатываний алгоритма. Каждый прямоугольник разбивается на два множества:  $P = \partial P \cup P^0$ , где  $\partial P$  — граница прямоугольника,  $P^0$  — внутренняя область,  $\partial P \cap P^0 = \emptyset$ , причем площадь  $S(P) = S(\partial P) + S(P^0)$ . Обозначение  $S(\partial P)$  соответствует длине периметра  $P$ : поскольку мы работаем с изображениями, состоящими из дискретного набора пикселей, то длиной периметра будет считаться количество пикселей граничного слоя рассматриваемого объекта.

Поиск прямоугольных строений с использованием преобразования Хафа на основе найденных алгоритмом Канны контуров (аналогично прямым) приведет к неустойчивому обнаружению даже строго прямоугольных строений из-за выбора шага при переборе возможных размеров прямоугольников (2): например, если перебирать нечетные длины сторон, в то время как контур образует прямоугольник со сторонами четной длины. Поэтому в предлагаемом алгоритме будут использоваться не контура, а модули градиентов (1): во-первых, в них нет разрывов, а во-вторых, за счет того, что они представляют собой достаточно широкую линию, то через них всегда можно провести прямой отрезок,

являющийся стороной некоторого прямоугольника. Более того, каждая точка будет голосовать за выбранный прямоугольник значением модуля градиента в ней, что позволит выбирать только те контура, которые соответствуют наибольшему суммарному перепаду яркости вдоль периметра прямоугольника.

Общая идея применения преобразования Хафа для поиска прямоугольных контуров изложена в [16]: предлагается на основе значений градиентов яркости построить для каждого набора параметров  $(w, h, \alpha)$  двумерный аккумуляторный массив, чей размер совпадает с размером исходного изображения, соответствующий точкам — центрам прямоугольников с выбранными параметрами. Далее строится результирующий аккумуляторный массив путем выбора в каждой точке максимального из значений построенных на предыдущем шаге аккумуляторов. Иными словами, мы рассматриваем каждую точку как потенциальный центр прямоугольника, перебираем все возможные прямоугольники, центром которых могла бы быть эта точка, и выбираем тот, который проходит через наибольшее (относительно длины периметра) количество граничных точек. Затем проводится пороговая обработка полученного результирующего аккумуляторного массива, выбираются локальные максимумы, и соответствующие им прямоугольники передаются на выход алгоритма. Необходимость выбора локальных максимумов обусловлена тем, что реальные снимки содержат шум и иные искажения, и даже один отдельно стоящий дом порождает не один контур, а целую группу, как правило, около десятка, сгруппированных вокруг его истинного центра.

Такой подход обладает очевидным недостатком: на реальных изображениях присутствует большое количество контуров, многие из которых могут образовывать прямоугольную структуру и приводить к появлению ложных срабатываний алгоритма: например, участки дорог, пространства между двумя близкорасположенными домами, лесные массивы и т. д. Поэтому требуется дополнить изложенную выше идею некоторыми условиями  $C_1, C_2, C_3$  заполнения аккумуляторного массива, отсекающими контура, не соответствующие строениям, но тем не менее имеющими форму, близкую к прямоугольной.

**Условие  $C_1$ : форма.** Данное условие учитывает близость формы рассматриваемых контуров к прямоугольнику. Используемые для построения аккумуляторного массива градиенты яркости позволяют не пропускать прямоугольники даже с размытыми границами. Однако, их использование порождает ложные тревоги, связанные с непрямоугольными объектами, которые, благодаря широкой полосе градиента, будут давать значительный вклад в аккумулятор. Поэтому необходим более строгий учет формы обнаруженного преобразованием Хафа прямоугольника  $P_{w,h,\alpha}$ , для чего будут использоваться контура, найденные алгоритмом Канни. Чтобы избежать пропуска отдельных точек контуров, которые относятся к рассматриваемому прямоугольнику, но из-за шума оказываются смещены относительно его периметра, будем рассматривать не только точки  $t$  периметра прямоугольника, но и их окрестность  $O(t)$ . Если в  $O(t)$  есть хоть одна точка контура, то считаем, что в точке  $t$  периметра есть контур. Для этого определим следующее одномерное изображение на точках периметра прямоугольника  $\partial P_{w,h,\alpha}$ :

$$F(t) = \max_{v \in O(t)} Canny(v), \quad (3)$$

причем  $F(t) \in \{0, 1\}$ . Размер окрестности выбирается в зависимости от степени зашумленности изображения; в данной работе была выбрана четырехсвязная окрестность.

Далее к изображению  $F$  последовательно применяются операции *эрозии* ( $\ominus$ ):

$$F'(t) = (F \ominus Q_q)(t) = \min_{v \in Q_q(t)} F(v) \quad (4)$$

и *дилатации* ( $\oplus$ ):

$$F''(t) = (F' \oplus Q_q)(t) = \max_{v \in Q_q(t)} F'(v), \quad (5)$$

где  $Q_q$  — одномерный структурирующий элемент [3] размера  $q = \frac{h}{4}$ , по сути задающий на периметре одномерную окрестность точки  $t$  радиуса  $q$ . В теории морфологического анализа последовательность операций (4)–(5) называется *открытием*. Применение операции открытия к  $F(t)$  приведет к удалению последовательностей точек контура длины менее, чем четверть длины меньшей из сторон прямоугольника. Такой подход позволяет учитывать лишь те контура, которые «сонаправлены» сторонам рассматриваемого прямоугольника, при этом отбрасывая слишком короткие отрезки контуров (которые, скорее всего, перпендикулярны сторонам прямоугольника).

После этого вычисляется сумма значений функции  $F''(\cdot)$  и выполняется проверка, что эта сумма не меньше, чем некоторый порог, в качестве которого в данной работе была выбрана половина длины периметра:

$$\sum_{t \in \partial P_{w,h,\alpha}(x,y)} F''(t) \geq \frac{S(\partial P_{w,h,\alpha})}{2}. \quad (6)$$

**Условие  $C_2$ : плотность границ.** При работе алгоритма, основанного на контурном анализе, значительную часть ложных тревог порождают сложнотекстурированные области, содержащие большое количество границ всевозможных направлений (характерным примером могут служить лесные массивы). В таких областях до трети точек могут быть отмечены алгоритмом Канни как содержащие контур, и, следовательно, в них можно найти большое количество ложных прямоугольников. На рис. 1 представлено два примера искомых строений с однородной внутренней областью ( $a$  и  $b$ ) и контуры, ограничивающие область со сложной структурой ( $c$  и  $d$ ).

Если рассматривать исключительно точки периметра прямоугольника, обе области содержат несколько искомых контуров, но можно отметить одно различие между ними: поскольку снимки инфракрасного диапазона отличаются нечувствительностью к мелким объектам, то крыша дома представляет собой достаточно однородную область, внутри которой мало контуров. Это наблюдение может служить еще одним признаком: во внутренней области рассматриваемого прямоугольника плотность границ должна быть незначительна (мы рассматривали значение в 5% от площади внутренней области). Данное условие можно записать следующим образом:

$$\sum_{t \in P_{w,h,\alpha}^0(x,y)} Canny(t) \leq \frac{S(P_{w,h,\alpha}^0)}{20}. \quad (7)$$

**Условие  $C_3$ : средняя яркость.** Особенностью спутниковых снимков является большое количество контуров в областях растительности, что может породить значительное количество ложных тревог. В то же время инфракрасные изображения характеризуются тем, что яркость изображений строений на них в среднем ниже, чем яркость окружающей растительности (на этом наблюдении основаны вегетационные индексы, в частности,

широкоизвестный NDVI [17]). Подобная информация никак не связана непосредственно с контурами, а потому хорошо их дополняет, позволяя отбрасывать однородные прямоугольные участки типа полей, газонов и т. д. Предлагаемое условие имеет следующий вид:

$$\frac{\sum_{t \in P_{w,h,\alpha}^0(x,y)} Image(t)}{S(P_{w,h,\alpha}^0)} < \frac{\sum_{t \in P_{w+r+d,h+r+d,\alpha}(x,y)} Image(t) - \sum_{t \in P_{w+r,h+r,\alpha}(x,y)} Image(t)}{S(P_{w+r+d,h+r+d,\alpha}) - S(P_{w+r,h+r,\alpha})}. \quad (8)$$

Здесь средняя яркость внутренней области прямоугольника ( $P^0$ ) сравнивается со средней яркостью прямоугольного «кольца» шириной  $d$ , построенного вокруг него на расстоянии  $r$ . Это расстояние представляет собой отступ от края прямоугольника: границы на реальных изображениях крайне редко имеют вид ступеньки, зато гораздо чаще они оказываются размыты на несколько пикселей (именно это размытие находит оператор Собела как градиент яркости), вследствие чего ближайšie к границе прямоугольника точки будут иметь практически такую же яркость, что и сам объект, а фон будет начинаться на несколько пикселей далее. Величина отступа зависит от типа и качества изображения, а ширина кольца  $d$  выбиралась так, чтобы его площадь была примерно равна площади внутренней области прямоугольника.

Таким образом, с учетом предложенных условий решающее правило алгоритма, определяющее наличие в некоторой произвольной точке  $(x, y)$  прямоугольника с параметрами  $(w, h, \alpha)$ , выглядит следующим образом:

$$P(x, y) = \begin{cases} P_{w,h,\alpha}(x, y), & \text{если } Acc(x, y) = \max_{(x',y') \in O_w(x,y)} Acc(x', y') \\ & \text{и выполнены } C_2, C_3, \\ \emptyset & \text{иначе,} \end{cases} \quad (9)$$

где

$$Acc(x, y) = \max_{w,h,\alpha} \left( \left( \frac{1}{w+h} \sum_{t \in \partial P_{w,h,\alpha}(x,y)} Sobel(t) \right) | C_1 \right), \quad (10)$$

$$P_{w,h,\alpha}(x, y) = \arg \max_{w,h,\alpha} \left( \left( \frac{1}{w+h} \sum_{t \in \partial P_{w,h,\alpha}(x,y)} Sobel(t) \right) | C_1 \right), \quad (11)$$

$O_w(x, y)$  — окрестность точки  $(x, y)$  радиуса  $w$ , т. е. совпадающего с максимальной по длине стороне прямоугольника, что позволяет отсесть случаи пересечения расположенных рядом прямоугольников;  $C_1, C_2, C_3$  — специальные условия, отражающие характерные свойства искомых объектов.

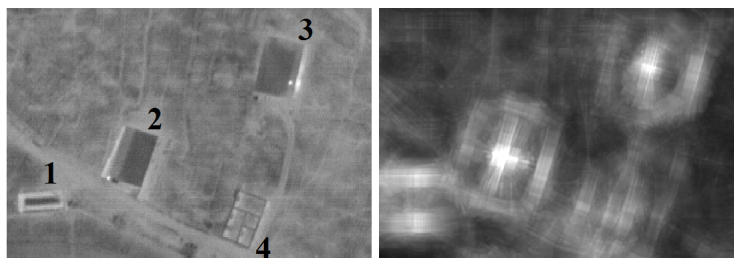
Завершим данный пункт описанием общей схемы предлагаемого алгоритма, состоящего из следующих шагов.

1. Сглаживание входного изображения (*guided filter*), вычисление градиентов яркости оператором Собела и поиск контуров алгоритмом Канни.

2. Для каждого набора параметров  $(w, h, \alpha)$  из выбранного диапазона значений (2) с помощью процедуры голосования точек градиентов яркости изображения (10) и условием совпадения формы  $C_1$  (6) строится аккумуляторный массив (размером, совпадающим с размером входного изображения, и соответствующий центрам прямоугольников). Затем

полученные аккумуляторные массивы объединяются операцией поточечного максимума с сохранением набора параметров (11), на котором этот максимум достигается.

3. В результирующем аккумуляторном массиве отбрасываются точки, не удовлетворяющие условиям плотности границ (7) и средней яркости (8). Среди оставшихся ищутся локальные максимумы (9) по окрестности размером с большую сторону прямоугольника  $w$ , которые передаются на выход алгоритма.



**Рис. 3.** Пример заполнения аккумуляторного массива при поиске прямоугольных контуров с помощью преобразования Хафа. Яркость соответствует значению аккумулятора: чем она выше, тем больше точек контуров исходного изображения проголосовали за прямоугольник с центром в этой точке. Представлены примеры строений (2 и 3), и типичные случаи прямоугольных объектов, искомыми строениями не являющихся (1 и 4)

На рис. 3 представлен пример построенного аккумуляторного массива, полученного в ходе поиска прямоугольных строений на изображении загородной местности. Можно видеть, что два прямоугольных строения породили в аккумуляторном массиве два выраженных локальных максимума (две яркие точки), соответствующих центрам зданий 2 и 3. В то же время объект 1 в левой нижней части изображения, чья ширина оказалась меньше рассматриваемых параметров фигур, породил лишь две размытые полосы, не локализуемые в конкретный объект. Помимо этого, учет однородности яркости внутри рассматриваемого прямоугольника позволил отбросить прямоугольный объект 4, не являющийся строением.

В следующем разделе будет приведено обсуждение реализации данного алгоритма, а также результаты численных экспериментов на реальных данных.

### 3. Реализация алгоритма и результаты численных экспериментов

Предложенный в предыдущем пункте алгоритм выполняет перебор контуров заданной формы всевозможных размеров и ориентаций, которые могли бы быть на входном изображении. Разумеется, если у нас нет априорной информации о размерах искомым объектов (при поиске строений на аэрокосмических изображениях это обычная ситуация), то такой перебор может потребовать значительного количества времени. Поэтому, если говорить о практическом применении данного метода, то остро встает вопрос об оптимизации вычислений и ускорении работы алгоритма. Последняя задача в настоящее время решается с помощью использования высокопроизводительных многоядерных графических процессоров. В связи с этим было принято решение оптимизировать алгоритм для работы на видеокартах; в качестве архитектуры параллельных вычислений была выбрана технология CUDA [18].

Сформулируем следующие идеи по оптимизации вычислений алгоритма.

1. Поскольку *guided filter* и оператор Собела представляют собой оконные методы (свертку), то их реализация на видеокарте не представляет затруднений. Быстрая реализация алгоритма Канни была взята из библиотеки OpenCV [19].

2. Рекуррентное вычисление аккумуляторного массива для каждого набора параметров  $(w, h, \alpha)$  позволяет заметно сократить объем требуемой памяти, если хранить только результирующий аккумуляторный массив, по мере необходимости обновляя его на каждой итерации. Также за счет того, что этот массив получается двумерным, в нем быстрее осуществляется поиск локальных максимумов.

3. Прореживание точек: снимки земной поверхности неизбежно содержат некоторое размытие изображенных объектов и шум, из-за чего локализации контура со сдвигом на пиксель по горизонтали или вертикали будут практически идентичны. В связи с этим нет необходимости рассматривать каждый пиксель изображения как потенциальный центр фигуры, так как значения аккумуляторного массива в двух соседних точках будут мало отличаться друг от друга: они оба будут содержать одинаковые параметры фигуры и вполне корректно локализовать искомый объект. Поэтому возможно учитывать каждый второй или третий пиксель, и тем самым заметно сократить размер аккумуляторного массива.

4. Идея преобразования Хафа — «точка голосует за центр фигуры» — может замедлить выполнение алгоритма на видеокарте, так как за этот центр может одновременно голосовать множество точек. Поэтому предлагается иной подход — «центр контура опрашивает все точки, которые могли бы за него проголосовать». Раз форма известна, то можно проверить все пиксели, которые могли бы проголосовать за фигуру с центром в выбранной точке, тем самым избегая возможной записи в одну и ту же ячейку памяти из разных потоков вычислений.

5. Предварительный расчет векторов сдвигов: при фиксированных параметрах фигуры каждый центр опрашивает точки с одним и тем же смещением относительно себя, поэтому можно предварительно рассчитать вектора сдвигов всех точек периметра фигуры, и далее использовать их для вычисления значения аккумулятора в каждой точке. Расчет поворота требует вычисления синусов и косинусов, и делать его для каждой точки в отдельности крайне затратно. Предварительный расчет векторов сдвигов позволяет выполнить эту операцию единожды, и затем вычислять координаты точек повернутой фигуры с помощью операций сложений и вычитаний.

6. Заполнение аккумуляторного массива (10) и проверка условия близости формы  $C_1$  (6) производятся по периметру фигуры, т. е. для прямоугольников, например, это порядка  $2(w + h)$  точек. В то же время условия  $C_2$  (7) и  $C_3$  (8) рассчитываются по внутренней области фигуры, что требует перебора порядка  $w \times h$  точек, а, следовательно, — значительно большего времени работы. Именно поэтому в (10) и (11) в качестве условий фигурирует лишь  $C_1$ , а два других условия вынесены за пределы заполнения аккумулятора (т. е. за пределы параллельной части) и проверяются только для точек предполагаемых центров прямоугольников (9). Добавление условий  $C_2$  и  $C_3$  в параллельную часть сильно замедлит работу алгоритма, но не приведет к заметному повышению качества работы.

Построенный на основе этих предложений алгоритм с перебором прямоугольников с размерами от  $l_{min} = 31$  до  $l_{max} = 91$  с шагом 2 (2), продемонстрировал время работы порядка двадцати двух секунд для изображения  $3000 \times 3000$  пикселей на системе, оснащенной процессором Intel i7-4770 и видеокартой Nvidia GeForce GTX 780 Ti.

Для проверки качества работы предложенного алгоритма использовалась база из 100 инфракрасных аэроснимков загородной местности размером  $3000 \times 3000$  пикселей, для которых была сделана ручная разметка строений прямоугольной формы. В случае, если алгоритм находил прямоугольный контур в небольшой окрестности от размеченного вручную и их размеры не сильно отличались, то такой контур считался верным обнаружением. Полученные результаты приведены в таблице.

Таблица

Результаты численного эксперимента по локализации строений прямоугольной формы

Характеристика	Количество	Процент
Строений найдено	2 213	78,45%
Строений пропущено	608	21,55%
Ложных тревог	410	15,63%

На рис. 4 приведены примеры работы алгоритма. В случае разреженного расположения объектов и относительно однородного фона алгоритм достаточно корректно обнаруживает искомые объекты, не допуская ошибок. Для более сложных сцен, характеризующихся плотной и хаотичной застройкой, алгоритм может пропускать некоторые объекты: либо по причине сложной формы (примеры 2 и 6), либо из-за того, что строения расположены вплотную друг к другу (примеры 3 и 4). Ложные тревоги связаны с объектами, подобно прямоугольным строениям также имеющим две параллельные и достаточно протяженные границы: дороги, мосты, участки берега и т. д. (примеры 1, 5 и 7). В то же время стоит отметить, что большую часть объектов алгоритм обнаружил корректно.

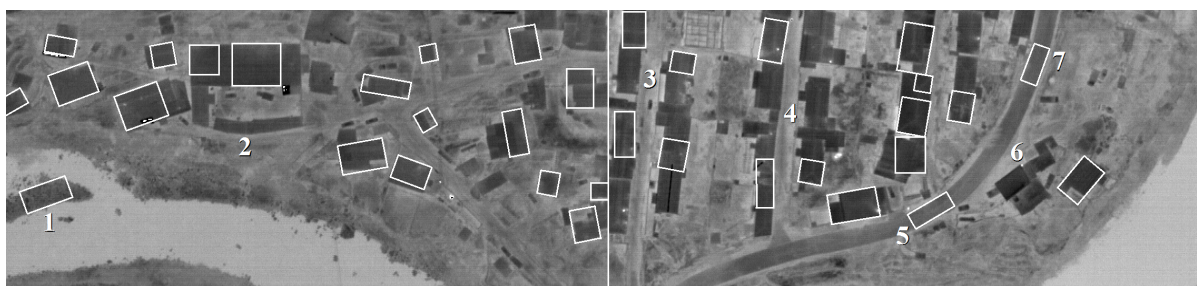


Рис. 4. Примеры работы алгоритма на аэроснимках плотной загородной застройки. Белым цветом выделены найденные строения. Цифрами 1, 5 и 7 отмечены ложные тревоги, цифрами 2, 3, 4 и 6 — пропуски

## Заключение

Предложенный в работе метод локализации строений заданной формы на аэроснимках в инфракрасном диапазоне применим к задаче поиска объектов различной формы, в общем случае, произвольных фигур, не обязательно выпуклых. При этом представляется, что чем сложнее искомая фигура, тем точнее будет работать алгоритм, но в то же время и значительно дольше. Однако стоит учитывать, что наиболее эффективную работу алгоритм сможет продемонстрировать для определенного класса изображений с особыми

спектральными и текстурными характеристиками: однородность фона, разреженное положение искомых объектов, разрешение порядка 1–3 метров на пиксель.

Помимо этого стоит учесть, что анализ контуров представляет собой достаточно неустойчивую к шуму и сложную в настройке операцию. Применение такого анализа в практических приложениях без использования дополнительных методов может быть затруднительно. Повышения универсальности можно добиться путем добавления элементов сегментации и анализа фона. Значительная часть ложных тревог связана с протяженными объектами типа дорог, чья ширина примерно соответствует ширине обычного дома. Предварительное разбиение снимка на однородные области поможет исключить такие ситуации. В свою очередь, уменьшение уровня ложных тревог позволит понизить пороги основного алгоритма и добиться более высокого уровня распознавания.

Также стоит упомянуть один из наиболее популярных в настоящее время подходов в обработке изображений, связанный с обучением многослойных сверточных нейронных сетей. Применение последних в основном связано с поиском объектов, обладающих достаточно сложной структурой: машин, животных и т.д. Строения на снимках в инфракрасном диапазоне, как было показано выше, подобной сложной структурой не обладают. Однако и для них можно подобрать некоторые характерные признаки. Причем некоторые из этих признаков сами по себе выражаются через операцию свертки: оператор Собела, плотность границ, средняя яркость области. Следовательно, приведенный в данной работе алгоритм может помочь при определении структуры нейронной сети, которая решала бы рассмотренную задачу.

*Работа выполнена при частичной поддержке комплексной программы ФНИ УрО РАН (проект 15-16-1-14).*

## Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Пер. с англ. М.: Техносфера, 2005. 1072 с.
2. Сирота А.А., Соломатин А.И. Статистические алгоритмы обнаружения границ объектов на изображениях // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2008. № 1. С. 58–64.
3. Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В., Ососков М.В., Моржин А.В. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. М.: Физматкнига, 2010. 672 с.
4. Леухин А.Н. Многомерный гиперкомплексный контурный анализ и его приложения к обработке изображений и сигналов. Диссертация на соискание ученой степени доктора физико-математических наук. Йошкар-Ола: Редакционно-издательский центр МарГТУ, 2004. 389 с.
5. Фурман Я.А. Введение в контурный анализ. 2-е изд., испр. М.: ФИЗМАТЛИТ, 2003. 592 с.
6. Hough P.V.C. Methods, Means for Recognizing Complex Patterns / U.S., Patent 3069654, 1962.
7. Heikkila M., Pietikainen M. A Texture-Based Method for Modeling the Background and Detecting Moving Objects // IEEE Transactions on Pattern Analysis and Machine



- Intelligence. 2006. Vol. 28, No. 4. pp. 657–662. DOI: 10.1109/TPAMI.2006.68.
8. Tomasi C., Manduchi R. Bilateral Filtering for Gray and Color Images // IEEE Proceedings of the 6-th International Conference on Computer Vision, Bombay, India. 7-th Jan. 1998. pp. 839–846. DOI: 10.1109/ICCV.1998.710815.
  9. K. He, J. Sun, X. Tang. Guided Image Filtering // IEEE Transactions on Software Engineering. 2013. Vol. 35(6). pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213.
  10. Duda R., Hart P. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973. pp. 271–272. DOI: 10.2307/1573081.
  11. Canny J. A Computational Approach to Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. PAMI-8, No. 6. pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
  12. L. Xu, E. Oja Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities // CVGIP: Image Understanding. 1993. Vol. 57, No. 2. pp. 131–154. DOI: 10.1006/ciun.1993.10091.
  13. Harris C., Stephens M. A Combined Corner and Edge Detector // Proceedings of the 4-th Alvey Vision Conference, University of Manchester. 31-st Aug. — 2-nd Sept. 1988. pp. 147–151. DOI: 10.5244/C.2.23.
  14. Rosten E., Porter R., Drummond T. Faster and Better: A Machine Learning Approach to Corner Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol. 32, No. 1. pp. 105–119. DOI: 10.1109/CIT.2010.109.
  15. Noronha S., Nevatia R. Detection and Modeling of Buildings from Multiple Aerial Images // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. Vol. 23, No. 5. pp. 501–518. DOI: 10.1109/34.922708.
  16. Корнилов Ф.А. Поиск прямоугольных контуров на изображениях с помощью преобразования Хафа // Современные проблемы математики и ее приложений: Труды 45-й международной молодежной школы-конференции (Екатеринбург, 2–8 февраля 2014 г.). Екатеринбург: ИММ УрО РАН, 2014. С. 195–198.
  17. Вегетационные индексы: URL: <https://earthobservatory.nasa.gov/Features/MeasuringVegetation/> (дата обращения: 18.05.2017)
  18. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектурная и программная модель CUDA. М.: Изд-во Моск. ун-та. Серия: Суперкомпьютерное образование, 2012. 336 с.
  19. Библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом: URL: <http://www.opencv.org/> (дата обращения: 18.05.2017)

Дунаева Александра Валерьевна, математик, отдел прикладных проблем управления, ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН (Екатеринбург, Российская Федерация); ассистент, кафедра высокопроизводительных компьютерных технологий, Институт естественных наук и математики, ФГБОУ ВО Уральский федеральный университет им. первого Президента России Б.Н. Ельцина (Екатеринбург, Российская Федерация)

Корнилов Фёдор Андреевич, к.ф.-м.н., н.с., отдел прикладных проблем управления, ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН (Екатеринбург, Российская Федерация)

---

DOI: 10.14529/cmse170306

## SPECIFIC SHAPE BUILDING DETECTION FROM AERIAL IMAGERY IN INFRARED RANGE

© 2017 A.V. Dunaeva<sup>1,2</sup>, F.A. Kornilov<sup>1</sup>

<sup>1</sup>*N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences (S.Kovalevskaya Str. 16, Yekaterinburg, 620990 Russia),*

<sup>2</sup>*Ural Federal University named after the first President of Russia B.N. Yeltsin (Mira Str. 19, Yekaterinburg, 620002 Russia)*

*E-mail: maryanova.av@yandex.ru, kornilovfa@imm.uran.ru*

Received: 24.07.2017

This paper describes an approach to detection of specific shape buildings from the aerial imagery in the infrared range. The proposed algorithm uses contour analysis and is based on a modification of the generalized Hough transform that allows to detect curves defined by a small number of parameters. The main idea is to build a two-dimensional accumulator array for each possible parameter set of the specified curve, and to combine obtained arrays into the resultant accumulator array whose local maxima correspond to the positions of the sought objects. The gradient magnitudes of the original image are used to fill the arrays. The closeness of the found contours to the predefined curve is determined by the morphological analysis of the values calculated by the Canny edge detector. Filtering detected objects relies on the density of boundaries in their internal area with the ratio of the average intensity inside and outside the contour that provides high sensitivity to the specified types of objects and reduces the number of false alarms of the algorithm. The proposed approach was tested on the problem of localization of rectangular buildings and showed the appropriate quality for practical use.

*Keywords: image processing, object detection, contour analysis, mathematical morphology.*

### FOR CITATION

Dunaeva A.V., Kornilov F.A. Specific Shape Building Detection from Aerial Imagery in Infrared Range. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 84–100. (in Russian) DOI: 10.14529/cmse170306.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

### References

1. Gonzalez R.C., Woods R.E. Digital Image Processing. 2-nd Edition, 2001. 1072 p.
2. Sirota A.A., Solomatin A.I. Statistical and Neural Network Algorithms of Allocation of Objects Border in Images. Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies. 2008. no. 1. pp. 58–64. (in Russian)
3. Vizilter Y.V., Zheltov S.Y., Bondarenko A.V., Ososkov M.V., Morzhin A.V. Image Processing and Analysis in Computer Vision: A Course of Lectures and Practical Exercises. Moscow, Publishing Fizmatkniga, 2010. 672 p. (in Russian)

4. Leukhin A.N. Multidimensional Hypercomplex Contour Analysis and Its Applications to Image and Signal Processing. Thesis for the Degree of Doctor of of Physico-Mathematical Sciences. Yoshkar-Ola, Publishing Mari State Technical University, 2004. 389 p. (in Russian)
5. Furman Y.A. Introduction to Contour Analysis. 2-nd Edition. Moscow, Publishing FIZMATLIT, 2003. 592 p. (in Russian)
6. Hough P.V.C. Methods, Means for Recognizing Complex Patterns. U.S., Patent 3069654, 1962.
7. Heikkila M., Pietikainen M. A Texture-Based Method for Modeling the Background and Detecting Moving Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006. vol. 28, no. 4. pp. 657–662. DOI: 10.1109/TPAMI.2006.68.
8. Tomasi C., Manduchi R. Bilateral Filtering for Gray and Color Images. IEEE Proceedings of the 6-th International Conference on Computer Vision, Bombay, India. 7-th Jan. 1998. pp. 839–846. DOI: 10.1109/ICCV.1998.710815.
9. K. He, J. Sun, X. Tang. Guided Image Filtering. IEEE Transactions on Software Engineering. 2013. vol. 35(6). pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213.
10. Duda R., Hart P. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973. pp. 271–272. DOI: 10.2307/1573081.
11. Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. vol. PAMI-8, no. 6. pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
12. L. Xu, E. Oja Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. CVGIP: Image Understanding. 1993. vol. 57, no. 2. pp. 131–154. DOI: 10.1006/ciun.1993.10091.
13. Harris C., Stephens M. A Combined Corner and Edge Detector. Proceedings of the 4-th Alvey Vision Conference, University of Manchester. 31-st Aug. — 2-nd Sept. 1988. pp. 147–151. DOI: 10.5244/C.2.23.
14. Rosten E., Porter R., Drummond T. Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. vol. 32, no. 1. pp. 105–119. DOI: 10.1109/CIT.2010.109.
15. Noronha S., Nevatia R. Detection and Modeling of Buildings from Multiple Aerial Images. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. vol. 23, no. 5. pp. 501–518. DOI: 10.1109/34.922708.
16. Kornilov F.A. Using the Hough Transform to Detect Rectangular Contours on Images. *Sovremennye problemy matematiki i ee prilozhenii: Trudy 45-i mezhdunarodnoi molodezhnoi shkoly-konferentsii (Yekaterinburg, 2 –8 fevralya 2014)* [Topical problems of mathematics and its applications: Proceedings of International (45-th National) Youth School-Conference (Yekaterinburg, Russia, February, 2–8, 2014)]. Yekaterinburg, Publishing IMM UB RAS, 2014. pp. 195–198. (in Russian)
17. Vegetation indices: URL: <http://go-url.ru/ix9o> (accessed: 18.05.2017)
18. Boreskov A.V. et. al. Parallel Computing on GPU. Architectural and Software Model CUDA. Moscow, Publishing Lomonosov Moscow State University. Series: Supercomputer education, 2012. 336 p. (in Russian)

19. The library of computer vision, image processing and computational mathematics algorithms with opened source code: URL: <http://www.opencv.org/> (accessed: 18.05.2017)

## СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по следующим отраслям и группам специальностей: 05.13.00 – информатика, вычислительная техника и управление; 01.01.00 – математика; 25.00.00 – науки о Земле (№421).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год (февраль, май, август и ноябрь).

## ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии <http://vestnikvmi.susu.ru>. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.**
2. Адрес редакции научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:  
Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, кафедра СП,  
ответственному секретарю Цымблеру М.Л.
3. Адрес электронной почты редакции: [vestnikvmi@susu.ru](mailto:vestnikvmi@susu.ru)
4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**