

## РАЗРАБОТКА АЛГОРИТМА ОПТИМИЗАЦИИ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ ПРИ ОПРЕДЕЛЕНИИ КОЛИЧЕСТВА НЕЙРОНОВ В СКРЫТОМ СЛОЕ В ЦЕЛЯХ ПОВЫШЕНИЯ ВЕРОЯТНОСТИ РАСПОЗНАВАНИЯ ОБРАЗОВ НАЗЕМНОЙ ЦЕЛИ

А. И. Годунов<sup>1</sup>, С. В. Шишков<sup>2</sup>, С. Т. Балаян<sup>3</sup>, Ф. Х. Аль Сафтли<sup>4</sup>

<sup>1</sup> Пензенский государственный университет, Пенза, Россия

<sup>2</sup> Филиал Военной академии материально-технического обеспечения имени генерала армии А. В. Хрулева в г. Пензе, Пенза, Россия

<sup>3,4</sup> Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина, Воронеж, Россия

<sup>1</sup> Godunov@pnzgu.ru, <sup>2</sup> sergej.shishkov.75@mail.ru, <sup>3</sup> bst76@yandex.ru, <sup>4</sup> fadi79@yandex.ru

**Аннотация.** *Актуальность и цели.* Высокая точность распознавания типовых наземных объектов оптико-электронными следящими системами может быть достигнута за счет оптимизации параметров искусственной нейронной сети (ИНС) такими, как: размерность и структура входного сигнала ИНС, синапсы нейронов сети, количество нейронов каждого слоя сети и количество слоев сети. *Материалы и методы.* Рассматриваются существующие алгоритмы оптимизации обучения ИНС при определении количества нейронов во входном, скрытом и выходном слоях ИНС в целях повышения вероятности распознавания образов наземной цели. Исследуются факторы улучшения обучения ИНС, определения количества нейронов в скрытом слое для распознавания изображений наземных объектов в таких алгоритмах, как алгоритм Левенберга – Марквардта, алгоритм байесовской регуляризации, масштабируемом алгоритме сопряженных градиентов и в разработанном алгоритме. *Результаты и выводы.* Исследуется возможность использования разработанного алгоритма в подсистеме информации и управления ракетой при телевизионном самонаведении на цель. Осуществлена программная реализация разработанного алгоритма с использованием языка программирования Matlab.

**Ключевые слова:** оптимизация, нейронная сеть, скрытый слой, обучение нейронной сети, алгоритм Левенберга – Марквардта, алгоритм байесовской регуляризации, масштабируемый алгоритм сопряженных градиентов, распознавание, вероятность, цель

**Для цитирования:** Годунов А. И., Шишков С. В., Балаян С. Т., Аль Сафтли Ф. Х. Разработка алгоритма оптимизации обучения нейронной сети при определении количества нейронов в скрытом слое в целях повышения вероятности распознавания образов наземной цели. 2021. № 4. С. 106–118. doi:10.21685/2307-4205-2021-4-12

## DEVELOPMENT OF AN ALGORITHM FOR OPTIMIZING NEURAL NETWORK TRAINING WHEN DETERMINING THE NUMBER OF NEURONS IN A HIDDEN LAYER IN ORDER TO INCREASE THE PROBABILITY OF RECOGNIZING IMAGES OF A GROUND TARGET

A.I. Godunov<sup>1</sup>, S.V. Shishkov<sup>2</sup>, S.T. Balanyan<sup>3</sup>, F.Kh. Al' Safftli<sup>4</sup>

<sup>1</sup> Penza State University, Penza, Russia

<sup>2</sup> Branch of the Military Academy of Logistics named after Army General A.V. Khrulev in Penza, Penza, Russia

<sup>3,4</sup> Air Force Academy named after Professor N. E. Zhukovsky and Yu. A. Gagarin, Voronezh, Russia

<sup>1</sup> Godunov@pnzgu.ru, <sup>2</sup> sergej.shishkov.75@mail.ru, <sup>3</sup> bst76@yandex.ru, <sup>4</sup> fadi79@yandex.ru

**Abstract.** *Background.* High accuracy of recognition of typical ground objects by optoelectronic tracking systems can be achieved by optimizing the parameters of an artificial neural network (INS) such as: the dimension and structure of the INS input signal, synapses of network neurons, the number of neurons of each network layer and the number of network layers. *Materials and methods.* The existing algorithms for optimizing the training of the INS are considered when determining the number of neurons in the input, hidden and output layers of the INS in order to increase the probability of recognizing images of a ground target. The factors of improving the training of the INS, determining the number of neurons in the hidden layer for recognizing images of ground objects in such algorithms as the Levenberg – Marquardt algorithm, the Bayesian regularization algorithm, the scalable conjugate

gradient algorithm and the developed algorithm are investigated. *Results and conclusions.* The possibility of using the developed algorithm in the subsystem of information and missile control during television homing on the target is investigated. The software implementation of the developed algorithm using the Matlab programming language is carried out.

**Keywords:** optimization, neural network, hidden layer, neural network training, Levenberg – Marquardt algorithm, Bayesian regularization algorithm, scalable conjugate gradient algorithm, recognition, probability, goal

**For citation:** Godunov A.I., Shishkov S.V., Balanyan S.T., Al' Safli F.Kh. Development of an algorithm for optimizing neural network training when determining the number of neurons in a hidden layer in order to increase the probability of recognizing images of a ground target. *Nadezhnost' i kachestvo slozhnykh sistem = Reliability and quality of complex systems.* 2021;(4):106–118. (In Russ.). doi:10.21685/2307-4205-2021-4-12

## Введение

В теории искусственных нейронных сетей (ИНС) существуют две актуальных проблемы, одной из которых является выбор оптимальной структуры нейронной сети, а другой – построение эффективного алгоритма обучения нейронной сети.

Оптимизация нейронной сети направлена на уменьшение объема вычислений при условии сохранения точности решения задачи на требуемом уровне. Параметрами оптимизации в нейронной сети могут быть:

- размерность и структура входного сигнала нейросети;
- синапсы нейронов сети. Они упрощаются с помощью удаления из сети или заданием «нужной» или «оптимальной» величины веса синапса;
- количество нейронов каждого слоя сети;
- количество слоев сети.

Вторая проблема заключается в разработке качественных алгоритмов обучения нейросети, позволяющих за минимальное время настроить нейросеть на распознавание заданного набора входных образов.

Обучение нейронной сети представляет собой многокритериальную задачу нелинейной оптимизации. Ее целью является поиск оптимального набора весовых коэффициентов (экстремума функции) для минимизации ошибки сети (рис. 1).

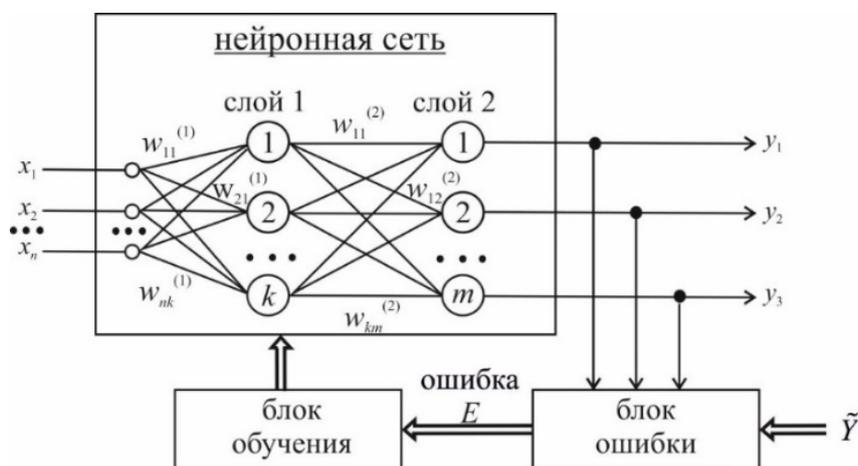


Рис. 1. Обучение многослойной нейронной сети

Процесс обучения нейронной сети заключается в необходимости настройки сети таким образом, чтобы для некоторого множества входов давать желаемое (или, по крайней мере, близкое, сообразное с ним) множество выходов.

## Выбор нейросетевой архитектуры

Создание архитектуры нейронной сети означает указание количества слоев каждого типа и количества нейронов в каждом из этих слоев [1].

Выбор нейросетевой архитектуры может быть сделан в зависимости от решаемой задачи, имеющихся исходных данных и планируемых алгоритмов обучения. Так, для решения задач клас-

сификации, аппроксимации и прогнозирования применимы сети прямого распространения [2]. Главным представителем сетей такого класса является многослойный персептрон, структура которого показана на рис. 2.

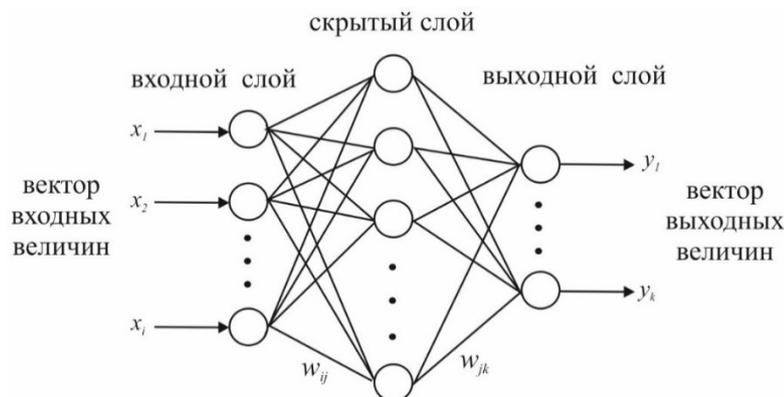


Рис. 2. Многослойный персептрон

В качестве исходных данных имеем [3]:

- размерность вектора входного сигнала ( $N_x$ );
- размерность вектора выходного сигнала ( $N_y$ );
- количество особых точек на изображении цели обучающей выборки ( $N_p$ ).

При этом выбираем:

- общее число нейронов в сети и число нейронов по слоям;
- функции активации нейронов;
- способ задания коэффициентов синоптической связи.

При построении нейронной сети накладываются следующие ограничения:

- число слоев в выходном слое равно числу классов;
- все входные сигналы подаются всем нейронам.

Количество нейронов входного слоя напрямую зависит от размерности исходного пространства входных данных (размерности выборки) [1]. Количество нейронов выходного слоя зависит от решаемой задачи и так же, как для входного слоя, от способов кодирования. Так, при решении задач анализа главных компонент и сжатия информации размерность выходного слоя берется точно такой же, как и входного. Для решения задач управления выходной слой обычно представлен одним нейроном. При классификации количество нейронов выходного слоя определяется методом кодирования тех классов, к которым может быть отнесен входной вектор [1].

### Количество нейронов скрытого слоя

Основной задачей при построении нейронной сети является определение числа скрытых слоев и числа нейронов в скрытых слоях, так как число нейронов во входном и выходном слоях определяется известным числом входов и выходов сети. Проблема выбора количества скрытых элементов многослойного персептрона заключается в том, что с одной стороны, число скрытых элементов должно быть достаточным для решения поставленной задачи, а с другой – не должно быть слишком большим, чтобы обеспечить необходимую обобщающую способность сети и избежать переобучения.

В общем, при решении задач распознавания в сети с одним скрытым слоем входной вектор преобразуется в некоторое новое пространство, которое может иметь другую размерность, а затем гиперплоскости, соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом, сеть распознает не только характеристики исходных данных, но и «характеристики характеристик», сформированные скрытым слоем. Сеть с одним скрытым слоем, содержащим нейроны со ступенчатой функцией активации, способна осуществить произвольную классификацию особых точек четырехмерного пространства. Более того, одного скрытого слоя нейронов с функцией активации достаточно для аппроксимации любой границы между классами или некоторой функции со сколь угодно высокой точностью [2].

Необходимое количество нейронов в скрытых слоях персептрона можно определить по формуле, являющейся следствием из теорем Арнольда – Колмогорова – Хехт – Нильсена. Однако эта

теорема не конструктивна, это лишь теорема существования, в ней не заложена методика определения числа нейронов в сети для некоторой конкретной обучающей выборки. Для первоначальной оценки числа нейронов в скрытых слоях однородных нейронных сетей часто используется формула для оценки необходимого числа синоптических весов  $N_w$  в многослойной сети [3]:

$$\frac{N_y Q}{1 + \log_2(Q)} \leq N_w \leq \left( \frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y, \quad (1)$$

где  $N_y$  – размерность выходного сигнала;  $Q$  – число элементов множества обучающих примеров;  $N_w$  – необходимое число синоптических связей;  $N_x$  – размерность входного сигнала.

Оценив с помощью этой формулы необходимое число синоптических связей  $N_w$ , можно рассчитать необходимое число нейронов в скрытых слоях. Например, число нейронов скрытого слоя ( $N_{ск}$ ) двухслойного персептрона будет равно

$$N_{ск} = \frac{N_w}{N_x + N_y}. \quad (2)$$

Для определения числа нейронов в скрытом или скрытых слоях существуют эмпирически обоснованные правила. Считается, что количество нейронов в скрытом слое должно быть в несколько раз меньше количества обучающих примеров при условии наличия избыточности обучающих данных.

Суммируя выше сказанное, для большинства задач можно было бы, вероятно, получить достойную производительность, установив конфигурацию скрытого слоя, используя только два правила:

- 1) количество скрытых слоев равно единице;
- 2) количество нейронов в этом слое является средним между количеством нейронов во входном и выходном слоях и быть в несколько раз меньше количества обучающих примеров при условии наличия избыточности обучающих данных [1]:

$$N_{ск} = \frac{N_x + N_y}{2}. \quad (3)$$

### Обучение нейронных сетей

Точность аппроксимации зависит от эффективности процедур обучения, поэтому при использовании нейронных сетей в прикладных задачах большое внимание уделяется алгоритмам, которые позволяют максимально точно «подстроить» весовые коэффициенты в зависимости от обучающей выборки [4].

Обучение нейронных сетей прямого распространения – это процесс определения значений весов сети на основе примеров, образующих обучающее множество для сети с  $n$  входами и  $m$  выходами, состоит из  $N$  входных и выходных значений – обучающих примеров. Структура обучающего множества имеет вид

$$\langle \tilde{X}, \tilde{Y} \rangle = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1, \overline{N}}, \quad (4)$$

где  $\tilde{x}_i \in R^n$  – входной вектор из  $i$ -го примера,  $\tilde{y}_i \in R^m$  – вектор желаемого выхода;  $R^n$  – множество входных сигналов;  $R^m$  – множество выходных сигналов.

Степень близости вектора-ответа нейросети  $y_i$  на  $i$ -м примере и соответствующего вектора желаемый выход  $\tilde{y}_i$  при текущем векторе весов нейронной сети  $w \in R^W$ , где  $W$  – количество весовых коэффициентов нейронной сети прямого распространения (НСПР), характеризуется мгновенным функционалом качества обучения ( $Q_i$ ) [5]:

$$Q_i = Q(\varepsilon_i(w)) = \varepsilon_i^T(w) \cdot V \cdot \varepsilon_i(w), \quad (5)$$

где  $\varepsilon_i(w) = y_i(w) - \tilde{y}_i$ ,  $\varepsilon_i \in R^m$  – вектор отклонений фактических выходов сети от желаемого выхода,  $V$  – определенная матрица, задающая взвешенную норму вектора  $\varepsilon_i(w)$ . Обычно  $V \in R^{m \times m}$  – единичная матрица, что сводит функционал к евклидовой норме вектора отклонений:

$$Q(\varepsilon_i(w)) = \varepsilon_i^T(w) \cdot \varepsilon_i(w) = (y_i(w) - \tilde{y}_i)^T (y_i(w) - \tilde{y}_i) = \sum_{i=1}^m (y_i(w) - \tilde{y}_i)^2. \quad (6)$$

Степень соответствия сети данным из обучающего множества задается интегральным функционалом качества обучения  $(E(w))$  [5]:

$$E(w) = \sum_{i=1}^N Q_i(w) = \sum_{i=1}^N \sum_{j=1}^m (y_i(w) - \tilde{y}_j)^2. \quad (7)$$

Для случая с одним выходом ( $m = 1$ ) и с учетом его обозначения, как  $F(x_i, w)$  – функция активации нейронов, принимает следующий вид:

$$E(w) = \sum_{i=1}^N Q_i(w) = \sum_{i=1}^N (y_i(w) - \tilde{y}_i)^2 = \sum_{i=1}^N (F(x_i, w) - \tilde{y}_i)^2. \quad (8)$$

Цель обучения НСПР – это определение такого вектора весов  $w^*$ , чтобы уравнение (8) принимало минимальное значение, что превращает процесс обучения сети в решение задачи безусловной оптимизации:

$$w^* = \arg \min_{w \in R^w} E(w). \quad (9)$$

Для решения (9) существует множество алгоритмов. На основе метода обратного распространения ошибки были основаны три алгоритма, которые могут быть использованы для обучения искусственной нейронной сети, которые, в свою очередь, были включены в программный пакет Matlab. Это алгоритм Левенберга – Марквардта, алгоритм байесовской регуляризации и масштабируемый алгоритм сопряженных градиентов.

**Алгоритм сопряженных градиентов.** В задаче обучения нейронной сети алгоритм сопряженных градиентов применяется для минимизации квадратичной функции вида  $f(w)$  [6]:

$$f(w) = \frac{1}{2} w^T G w - b^T w + c, \quad (10)$$

где  $w$  – количество весовых коэффициентов;  $G$  – матрица Гесссиана;  $b, c$  – постоянные значения.

Направление поиска экстремума выбирается так, чтобы оно было ортогональным и сопряженным со всеми предыдущими значениями направления градиента ( $p_k$ ):

$$p_k = -g_k + \beta_{k-1} p_{k-1}, \quad (11)$$

где  $g_k = g(w_k)$  – фактическое значение направления градиента;  $\beta_k$  – коэффициент, используемый для определения сопряженного направления.

Изменение весов рассчитывается по формуле

$$w_{k+1} = w_k + \alpha_k p_k. \quad (12)$$

Коэффициент  $\alpha_k$  используется для определения сопряженного направления. Значение  $\alpha_k$  должно минимизировать  $f(w_{k+1})$ , для расчета можно применить алгоритм золотого сечения или другой линейный алгоритм поиска минимума. Для расчета коэффициента  $\beta_k$  применяется формула Флетчера – Ривза:

$$\beta_k = \frac{g_k^T \cdot g_k}{g_{k-1}^T \cdot g_{k-1}}. \quad (13)$$

Для квадратичных функций алгоритм сопряженных градиентов находит минимум за  $n$  шагов ( $n$  – размерность пространства поиска), поэтому алгоритм нуждается в перезапуске на каждом  $n + 1$  шаге. Для метода Флетчера – Ривза это выполняется с помощью  $\beta_k = 0$ . Полный алгоритм выглядит следующим образом:

- 1) инициализация весов  $w_k (k = 1)$ ;
- 2) расчет градиента  $g_1 = \nabla f(w_1)$ ;

- 3) определение направления  $p_1 = -g_1$ ;
  - 4) расчет  $\alpha_k = \arg \min [f(w_k + \alpha_k p_k)]$ ;
  - 5) коррекция весовых коэффициентов  $w_{k+1} = w_k + \alpha_k p_k$ ;
  - 6) если ошибка ( $E$ ) незначительна – остановить обучение;
  - 7) если  $k + 1 > n$  установить  $w_1 = w_{k+1}$ ,  $k = 1$  и перейти к шагу 2;
  - 8) определить  $k = k + 1$ ;
  - 9) рассчитать  $g_{k+1} = \nabla f(w_{k+1})$ ,  $\hat{\alpha}_k$ , вектор направления  $p_{k+1} = -g_{k+1} + \beta_k p_k$  и перейти к шагу 4.
- Для расчета градиента на этапах 2 и 9 используется целевая функция ошибки нейронной сети:

$$f(w) = \frac{1}{2N} \sum_n \sum_j (y_{nj} - \tilde{y}_{nj})^2, \quad (14)$$

где  $N$  – количество обучающих образов;  $y_{nj}$  и  $\tilde{y}_{nj}$  – фактический и желаемый выходы сети соответственно.

### Алгоритм Левенберга – Марквардта

Для устранения указанных недостатков обычно используется информация высокого порядка об  $E(w)$ . В рамках квадратичной аппроксимации ошибки в окрестности точки  $w$  имеет вид [5]:

$$E(w + \Delta w) \approx E(w) + \nabla E(w) \Delta w + \frac{1}{2} \Delta w^T \nabla^2 E(w) \Delta w. \quad (15)$$

На основе квадратичной аппроксимации разработаны широко известные алгоритмы Гаусса – Ньютона и Левенберга – Марквардта (ЛМ-алгоритм), которые сводят задачу (9) для (15) к уравнению

$$\nabla E(w) + \nabla^2 E(w) \Delta(w) = 0, \quad (16)$$

$$\nabla E(w) = \frac{\partial \varepsilon(w)}{\partial w} \cdot \varepsilon(w) = \frac{\partial (y(w) - \tilde{y})}{\partial w} \cdot \varepsilon(w) = \frac{\partial F(x, w)}{\partial w} \cdot \varepsilon(w), \quad (17)$$

$$\nabla^2 E(w) = \left( \frac{\partial \varepsilon(w)}{\partial w} \right)^T \cdot \frac{\partial \varepsilon(w)}{\partial w} + \sum_{i=1}^N \varepsilon(w) \nabla^2 \varepsilon(w). \quad (18)$$

Ключевое различие между ними – подход к вычислению матрицы Гессе  $\nabla^2 E(w)$ . Если представить (18) в виде

$$H = J^T \cdot J + S, \quad (19)$$

где  $S$  – информация о вторых производных, то для алгоритма Гаусса – Ньютона  $S = 0$ , в то время как в ЛМ-алгоритме  $S$  аппроксимируется эвристическими правилами.

Исходя из (16)–(19), алгоритм Левенберга – Марквардта заключается в решении уравнения относительно  $\Delta w$  [5]:

$$\underbrace{(J^T J)}_{H^*} + \underbrace{\lambda I}_S \Delta(w) = J^T \varepsilon(w), \quad (20)$$

или в другой интерпретации

$$\begin{aligned} (J^T J + \lambda I) \delta &= J^T \tilde{\varepsilon}(w), \\ \tilde{\varepsilon}(w) &= -\varepsilon(w) = \tilde{y} - y(w), \end{aligned} \quad (21)$$

где  $\lambda$  – коэффициент затухания Левенберга;  $\delta$  – вектор, состоящий из величин приращения весов;  $I$  – единичная матрица.

Найденный вектор  $\delta$  позволяет изменить вектор весов  $w$ . Элементы вектора  $w$  обычно упорядочиваются сначала по слою, затем по нейронам и, наконец, по весу каждого нейрона и его смещению [5].

Параметр  $\lambda$  задается изначально и определяет поведение алгоритма, делая его более похожим на градиентный или алгоритм Гаусса – Ньютона. В самом начале обучения, когда функция  $F(x, w)$  подобрана грубо, удобно использовать алгоритм наискорейшего спуска, поэтому  $\lambda$  выбирается относительно большим. По мере уточнения коэффициентов  $w$  более эффективным становится алгоритм Гаусса – Ньютона (при этом  $\lambda$  становится малой величиной; при  $\lambda = 0$  метод вырождается в алгоритм Гаусса – Ньютона). Так, ЛМ-алгоритм реализует адаптивную модель с явной аппроксимацией  $S$  и без нее.

В результате процедура, реализующая обучающий цикл НСПР, имеет вид [5]:

- 1) построить матрицу Якоби  $J$ ;
- 2) рассчитать градиент ошибки  $g = J^T \tilde{\epsilon}(w)$ ;
- 3) рассчитать приближенную матрицу Гессе с помощью матрицы Якоби  $H^* = J^T J$ ;
- 4) решить уравнение  $(H^* + \lambda I)\delta = g$  относительно неизвестного вектора  $\delta$ ;
- 5) вычислить  $E(w + \delta)$ ;
- 6) если  $E(w + \delta) < E(w)$ , то  $\lambda = \nu\lambda$  и перейти на шаг 4, иначе  $\lambda := \lambda | \nu$ ,  $E(w + \delta) := E(w)$  и закончить цикл обучения.

### Алгоритм байесовской регуляризации обучения

В байесовском подходе выбор решения  $f$  ведется в рамках некоторой модели  $H$  – набора ограничивающих условий. Решение выбирается как наиболее вероятное по формуле Байеса [7]:

$$P(f | D, H) = \frac{P(D | f, H)P(f | H)}{P(D | H)}, \quad (22)$$

где  $P(f | D, H)$  – вероятность выбора функции  $f$  для описания данных  $D$  в рамках модели  $H$ ;  $P(D | f, H)$  – вероятность того, что функция  $f$  генерирует данные  $D$ ;  $P(f | H)$  – некоторая априорная вероятность выбора  $f$  в рамках модели  $H$ . Коэффициент  $P(D | H)$  необходим, чтобы нормировать множество значений произведения  $P(D | f, H)P(f | H)$  на единицу, придав ему, таким образом, смысл вероятности.

Насколько подходит тот или иной набор ограничивающих условий  $H$  для описания данных, определяет как раз знаменатель формулы Байеса [7]:

$$P(D | H) = \sum_f P(D | f, H)P(f | H) = \sum_f P(D, f | H). \quad (23)$$

В научно-технической литературе имеется термин *evidence* (доказательство). Чем больше *evidence*, тем соответствующая модель более предпочтительна [7].

Сделав несколько предположений о внешнем виде вероятностей в формуле (22), можно найти параметр регуляризации исходя из принципа максимального правдоподобия. Если предположить, что данные зашумлены гауссовым шумом, тогда вероятность порождения функцией  $f(x)$  данных  $D = \{y_i, \bar{x}_i\}_{i=1}^L$  можно оценить следующим выражением [7, 8]:

$$P(D | f) = \frac{1}{Z_x} \exp\left(-\beta \sum_{i=1}^L (y_i - f(\bar{x}_i))^2\right). \quad (24)$$

Возьмем стабилизирующий функционал в гауссовой форме:

$$\Omega = \sum_{n=1}^N w_n^2. \quad (25)$$

Ему соответствует априорная вероятность:

$$P(f | H) = \frac{1}{Z_A} \exp\left(-\alpha \sum_{n=1}^N w_n^2\right). \quad (26)$$

Максимизация вероятности:

$$P(f | D, H) = \frac{1}{Z_M} \exp(-M), \quad (27)$$

где  $M = \beta \sum_{i=1}^L (y_i - f_i)^2 + \alpha \sum_{n=1}^N w_n^2$  сводится к решению системы линейных алгебраических уравнений:

$$\sum w_n A_{mn} = B_n, \quad (28)$$

где  $A_{mn} = \beta \sum_{i=1}^L \psi_{mi} \psi_{ni}$ ,  $n \neq m$ ,  $A_{nn} = \beta \sum_{i=1}^L \psi_{ni}^2 + \alpha$ ,  $B_n = \beta \sum_{i=1}^L y_i \psi_{ni}$ ;  $\psi_{ni}$  – базисная функция;  $A_{nn}, A_{mn}, B_n$  – симметричные матрицы с положительными элементами главной диагонали;  $\alpha, \beta$  – параметры модели обучения.

В качестве значений  $\alpha$  и  $\beta$  берутся те, при которых *evidence* максимально. Таким образом, чтобы их определить, можно, например, получить аналитическое выражение для *evidence*, а затем градиентными методами найти его максимум. Легко показать, что

$$\ln P(D | H) = \ln Z_M - \ln Z_A - \ln Z_X, \quad (29)$$

$$Z_M = \int d^N w_n \times \exp \left( -\beta \sum_{n=1}^N \left( \bar{y} - \sum_{n=1}^N w_n \bar{\psi}_n \right)^2 - \alpha \sum_{n=1}^N w_n^2 \right),$$

$$Z_A = \int_{-\infty}^{+\infty} \exp \left( -\alpha \sum_{n=1}^N w_n^2 \right) = \left( \frac{2\pi}{\alpha} \right)^{\frac{N}{2}}, \quad Z_X = \left( \frac{\pi}{\beta} \right)^{\frac{L}{2}},$$

где  $Z_A, Z_M, Z_X$  – нормировочные коэффициенты; через  $\bar{\psi}_n$  обозначены векторы значений  $n$ -й базисной функции в точках  $\{\bar{x}_i\}_{i=1}^L$ , а через  $\bar{y}$  – вектор значений искомой функции в этих точках.

Подробности вычисления этого интеграла описаны в работе [8]. В результате приходим к следующему приближенному выражению для *evidence*:

$$\ln P(D | H) = \sum_{n=1}^N \left( \frac{(\beta \bar{y} \bar{\psi}_n)^2}{A_{nn}} - \ln A_{nn} - \beta \sum_{m \neq n} \bar{\psi}_n \bar{\psi}_m w_n w_m \right) + \frac{N}{2} \ln \alpha - \beta \bar{y} + \frac{L}{2} \ln \beta - \frac{L}{2} \ln \pi. \quad (30)$$

Данный функционал содержит коэффициенты разложения  $\{w_n\}_{n=1}^N$ , для нахождения которых из формулы (20), в свою очередь, необходимо знать значения параметров  $\alpha$  и  $\beta$ . В таких случаях обычно пользуются (Expectation Maximization / ожидание максимизация) EM-алгоритмом [7]:

– на первом шаге случайным образом фиксируют параметры модели  $\alpha$ ,  $\beta$  и вычисляют значения  $\{w_n\}_{n=1}^N$  из уравнения (28);

– на втором шаге находят новые параметры  $\alpha$  и  $\beta$  путем поиска максимума функционала (30) при текущих значениях коэффициентов разложения  $\{w_n\}_{n=1}^N$ .

Далее параметры модели, полученные на втором шаге, используют для получения новых коэффициентов разложения и т.д. При байесовском подходе к регуляризации задач обучения нейронной сети многократно повторяются эти шаги. Таким образом, алгоритм сводится к некоторому решению  $f(\bar{x})$  [7].

### Моделирование

Для проведения экспериментов была разработана программа в среде Matlab для обучения нейронной сети, состоящей из двух слоев (входной и скрытый) с использованием алгоритма Левенберга – Марквардта, алгоритма байесовской регуляризации, масштабируемого алгоритма сопряженных градиентов и разработанного алгоритма. Количество нейронов в скрытом слое определялось с использованием теории Арнольда – Колмогорова – Хехт – Нильсена (1) по критериям максимальной вероятности распознавания объекта и минимального времени обучения (рис. 3).

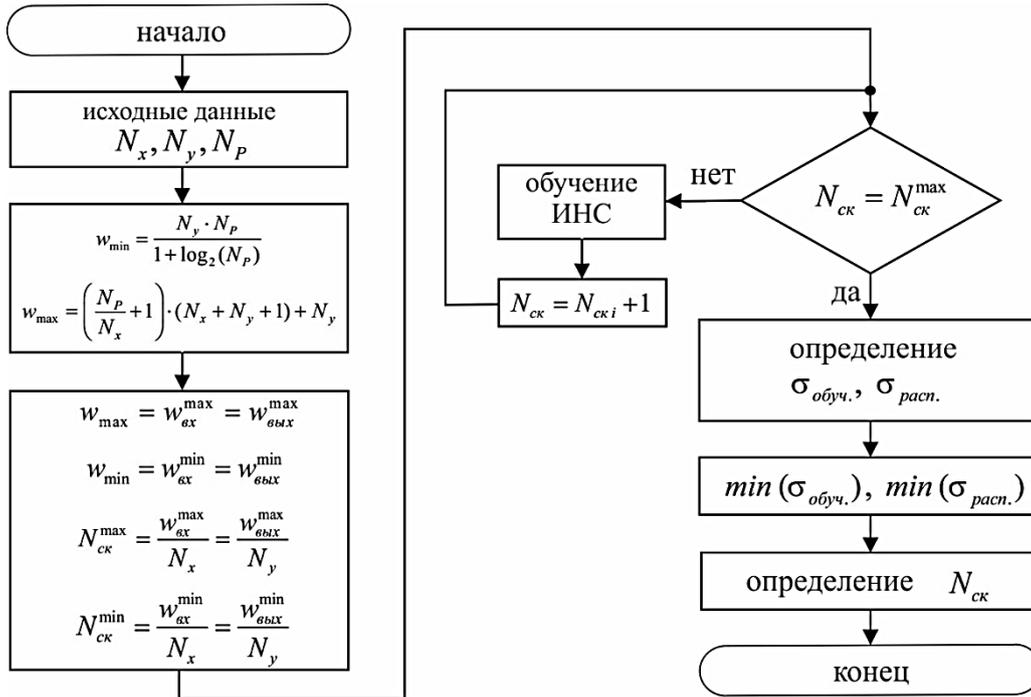


Рис. 3. Разработанный алгоритм оптимизации обучения нейронной сети при определении количества нейронов в скрытом слое

При этом  $N_p$  – количество особых точек на изображении цели [9] обучающей выборки ( $N_p = 2098$ );  $N_y$  – размерность выходного сигнала ( $N_y = 2$  jeep или tank);  $N_x$  – размерность входного сигнала ( $N_x = 64$ );  $w_{вх}^{max}$ ,  $w_{вых}^{max}$  – максимальное количество весов между входным и скрытым слоем, максимальное количество весов между скрытым и выходным слоем соответственно;  $w_{вх}^{min}$ ,  $w_{вых}^{min}$  – минимальное количество весов между входным и скрытым слоем, минимальное количество весов между скрытым и выходным слоем соответственно;  $w_{max}$ ,  $w_{min}$  – максимальное и минимальное количество весов соответственно;  $N_{ск}$  – количество нейронов в скрытом слое;  $T_{обуч}$  – время обучения;  $P_{расп}$  – вероятность распознавания;  $\sigma_{обуч}$  – ошибка обучения;  $\sigma_{расп}$  – ошибка распознавания.

Проведены исследования по определению количества нейронов в скрытом слое, вероятности распознавания объекта и времени обучения нейронной сети (рис. 4–6).

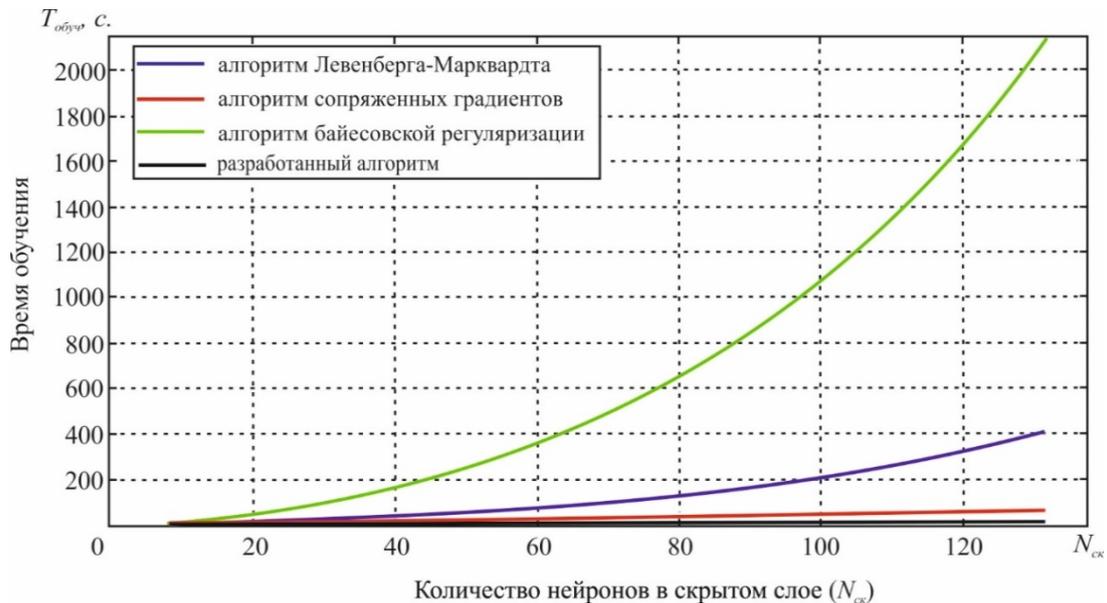


Рис. 4. Зависимость времени обучения ИНС от количества нейронов в скрытом слое ( $N_{ск}$ )

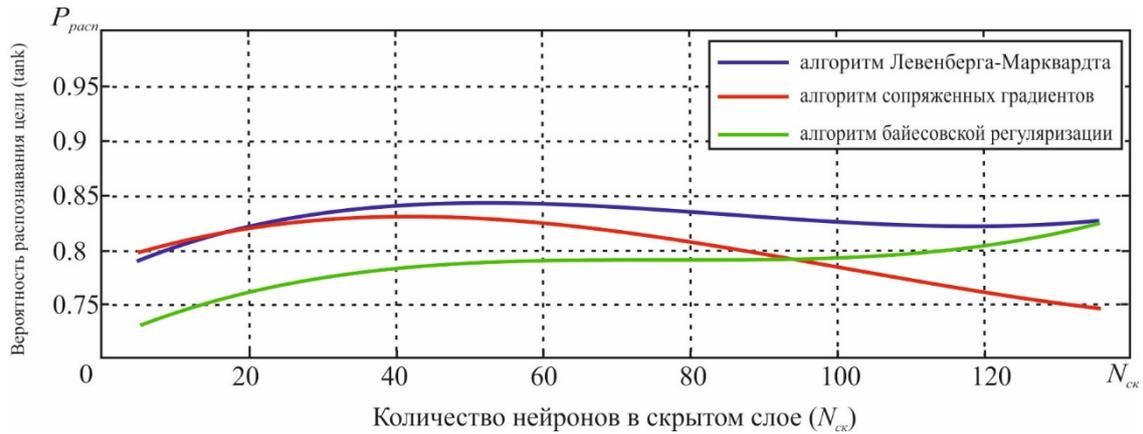


Рис. 5. Зависимость вероятности распознавания цели (tank) ИНС от количества нейронов в скрытом слое ( $N_{ск}$ )

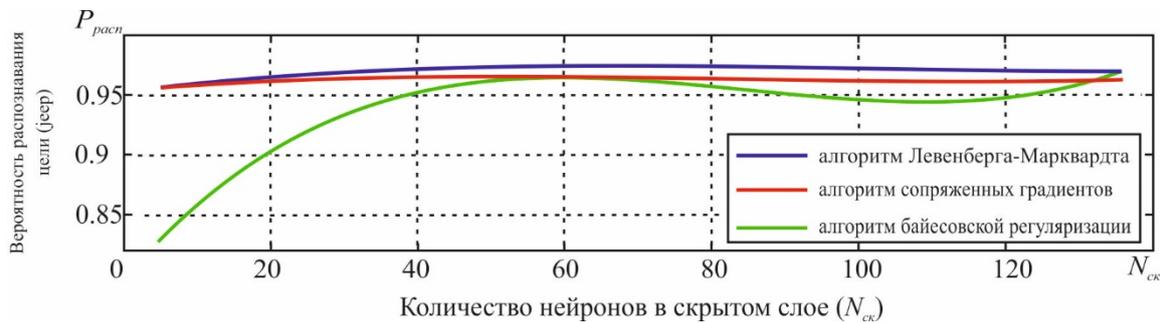


Рис. 6. Зависимость вероятности распознавания цели (jeep) ИНС от количества нейронов в скрытом слое ( $N_{ск}$ )

На рис. 4–6 показано, что время обучения многослойного персептрона нейронной сети для байесовского алгоритма к регуляризации обучения больше, чем у алгоритма Левенберга – Марквардта и масштабируемого алгоритма сопряженных градиентов, у которого время обучения меньше, чем у остальных. Вероятность распознавания цели (tank или jeep) у алгоритма Левенберга – Марквардта больше, чем у масштабируемого алгоритма сопряженных градиентов и байесовского алгоритма организации обучения.

Эмпирически определяется количество нейронов в скрытом слое с использованием уравнений (1), (3) и с помощью алгоритма Левенберга – Марквардта по критерию максимальной вероятности распознавания объекта (см. рис. 3). В результате оптимальное число нейронов в скрытом слое составило  $N_{ск} = 37$  с вероятностью распознавания объекта  $P_{расп} \approx 0,99$  (рис. 7, 8).

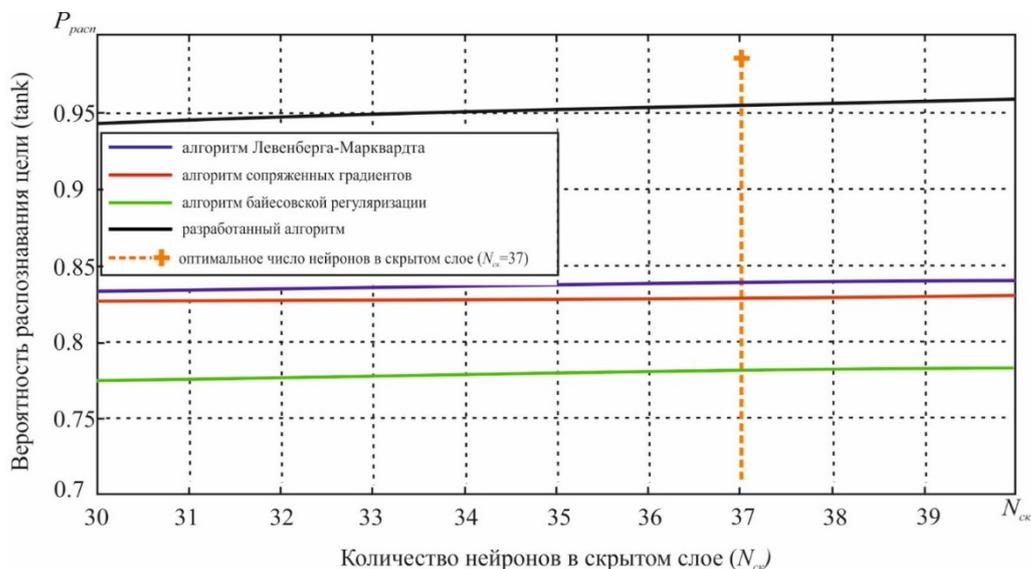


Рис. 7. Зависимость вероятности распознавания цели (tank) от количества нейронов в скрытом слое ( $N_{ск}$ )

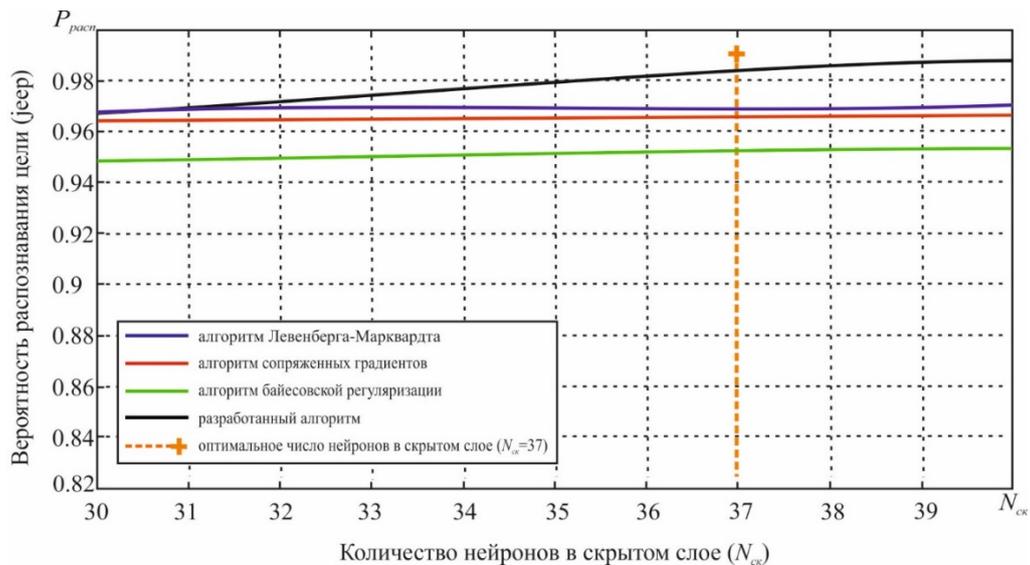


Рис. 8. Зависимость вероятности распознавания цели ( $P_{расп}$ ) от количества нейронов в скрытом слое ( $N_{ск}$ )

При использовании части или всех весов нейронной сети  $N_{ск} = 37$ , полученных эмпирически, для определения количества нейронов в скрытом слое с помощью алгоритма масштабируемых сопряженных градиентов для обучения сети по критерию максимальной вероятности распознавания объекта получаем достаточно высокую вероятность распознавания объекта (рис. 7, 8).

### Результаты

Процесс распознавания цели с помощью ИНС состоит из трех этапов. Первый – это извлечение признаков из эталонных изображений по алгоритму обнаружения особых точек SURF и формирование из них базы данных; второй этап – определение ( $N_{ск}$ ) и обучение нейронной сети с использованием этих данных и желаемых выходов  $\tilde{y}_i$ ; третий этап – это этап распознавания объекта, в процессе которого извлечены признаки из текущего изображения и применены в входе построения нейронной сети.

Особая точка на изображении цели, полученная с помощью алгоритма SURF, состоит из 64 градиентов дескрипторов относительно особой точки, поэтому количество нейронов во входном слое ( $N_x$ ) равно 64.

Количество нейронов в скрытом слое определялось экспериментально с использованием разработанного алгоритма по критерию максимальной вероятности распознавания объекта и в диапазоне количества нейронов в скрытом слое (30 ÷ 40) (см. рис. 7, 8). Таким образом, на этапе распознавания цели при использовании разработанной ИНС, в которой получено оптимальное число нейронов в скрытом слое ( $N_{ск} = 37$ ), имеем возможность ее обучения с целью получения высокой вероятности распознавания образов наземной цели  $P_{расп} \approx 0,99$  (см. рис. 7, 8).

### Список литературы

1. Прус А. И. Разработка нейронной сети прямого распространения для решения задач прогнозирования и аппроксимации : магистерская диссертация. СПб. : СПбГУ, 2016. 324 с.
2. Евдокимов И. А., Солодовников В. И. Автоматизация построения нейронной сети в рамках объектно-ориентированного подхода. М. : Центр информационных технологий в проектировании (ЦИТИ) РАН, 2015. С. 34–45.
3. Методы классической и современной теории автоматического управления : учебник в 5 т. / под ред. К. А. Пупкова, Н. Д. Егупова. 2-е изд., перераб. и доп. М. : Изд-во МГТУ им. Н. Э. Баумана, 2004. Т. 5: Методы современной теории автоматического управления. 784 с.
4. Пархоменко С.С. О сокращении времени обработки большого количества данных нейронными сетями методом Левенберга – Марквардта // Международный научно-исследовательский журнал. 2014. № 1. С. 123–134.
5. Пархоменко С. С., Леденева Т. М. Обучение нейронных сетей методом Левенберга-Марквардта в условиях большого количества данных // Вестник ВГУ. Сер.: Системный анализ и информационные технологии. 2014. № 2. С. 66–78.

6. Кучеренко А. Г. Обучение нейронной сети с использованием алгоритма сопряженных градиентов и множественной линейной регрессии для задачи прогнозирования спроса // ДГТУ экономические науки. Математические методы в экономике. 2009. № 1. С. 27–34.
7. Нужный А. С. Байесовский подход к регуляризации задачи обучения сети функций радиального базиса // Искусственный интеллект и принятие решений. 2015. № 1. С. 167–179.
8. Нужный А. С. Байесовская регуляризация в задаче аппроксимации функции по точкам с помощью ортогонализированного базиса // Физический институт им. П. Н. Лебедева РАН. Математическое моделирование. 2011. Т. 23, № 9. С. 33–42.
9. Годунов А. И., Балаян С. Т., Егоров П. С. Сегментация изображений и распознавание объектов на основе технологии сверточных нейронных сетей // Надежность и качество сложных систем. 2021. № 3. С. 62–73.

### References

1. Prus A.I. *Razrabotka neyronnoy seti pryamogo rasprostraneniya dlya resheniya zadach prognozirovaniya i aproksimatsii: masterskaya dissertatsiya = Development of a neural network of direct propagation for solving forecasting and approximation problems : master's thesis.* Saint Petersburg: SPBGU, 2016:324. (In Russ.)
2. Evdokimov I.A., Solodovnikov V.I. *Avtomatizatsiya postroeniya neyronnoy seti v ramkakh ob"ektно-orientirovannogo podkhoda = Automation of building a neural network in the context of object-oriented approach.* Moscow: Tsentr informatsionnykh tekhnologiy v proektirovanii (TsITI) RAN, 2015:34–45. (In Russ.)
3. Pupkov K.A., Egunov N.D. (ed.). *Metody klassicheskoy i sovremennoy teorii avtomaticheskogo upravleniya: uchebnik v 5 t. = The methods of classical and modern theory of automatic control : tutorial in 5 volumes.* 2nd ed., rev. and suppl. Moscow: Izd-vo MGTU im. N. E. Baumana, 2004;5:784. (In Russ.)
4. Parkhomenko S.S. On reducing the processing time of a large amount of data by neural networks using the Levenberg-Marquardt method. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal = International Research Journal.* 2014;(1):123–134. (In Russ.)
5. Parkhomenko S.S., Ledeneva T.M. Training of neural networks by the Levenberg-Marquardt method in conditions of a large amount of data. *Vestnik VGU. Ser.: Sistemy analiz i informatsionnye tekhnologii = Vestnik VSU. Ser.: System analysis and information technologies.* 2014;(2):66–78. (In Russ.)
6. Kucherenko A.G. Neural network training using the algorithm of conjugate gradients and multiple linear regression for the problem of demand forecasting. *DGTU ekonomicheskie nauki. Matematicheskie metody v ekonomike = DSTU ekonomicheskie nauki. Mathematical methods in economics.* 2009;(1):27–34. (In Russ.)
7. Nuzhnyy A.S. Bayesian approach to regularization of the problem of training a network of radial basis functions. *Iskusstvennyy intellekt i prinyatie resheniy = Artificial intelligence and decision-making.* 2015;(1):167–179. (In Russ.)
8. Nuzhnyy A.S. Bayesian regularization in the problem of function approximation by points using an orthonalized basis. *Fizicheskyy institut im. P. N. Lebedeva RAN. Matematicheskoe modelirovanie = Lebedev Physical Institute of the Russian Academy of Sciences. Mathematical modeling.* 2011;23(9):33–42. (In Russ.)
9. Godunov A.I., Balanyan S.T., Egorov P.S. Image segmentation and object recognition based on convolutional neural network technology. *Nadezhnost' i kachestvo slozhnykh system = Reliability and Quality of Complex Systems.* 2021;(3):62–73. (In Russ.)

### Информация об авторах / Information about the authors

#### Анатолий Иванович Годунов

доктор технических наук, профессор,  
 профессор кафедры автоматизации и телемеханики,  
 Пензенский государственный университет  
 (Россия, г. Пенза, ул. Красная, 40)  
 E-mail: Godunov@pnzgu.ru

#### Anatoly I. Godunov

Doctor of technical sciences, professor,  
 professor of the sub-department  
 of automatics and telemechanics,  
 Penza State University  
 (40 Krasnaya street, Penza, Russia)

#### Сергей Викторович Шишков

кандидат технических наук, доцент,  
 доцент кафедры производства и эксплуатации  
 ракетно-артиллерийского вооружения,  
 Филиал Военной академии материально-технического  
 обеспечения имени генерала армии  
 А. В. Хрулева в г. Пензе  
 (Россия, г. Пенза, Военный городок, 1)  
 E-mail: sergej.shishkov.75@mail.ru

#### Sergey V. Shishkov

Candidate of technical sciences, associate professor,  
 associate professor of the sub-department of production  
 and operation of missile and artillery weapons,  
 Branch of the Military Academy of Logistics  
 named after Army General  
 A.V. Khrulev in Penza  
 (1 Voennyuy gorodok, Penza, Russia)

**Сергей Товмасович Балабян**

кандидат технических наук, доцент,  
докторант кафедры авиационного вооружения  
и эффективности боевого применения,  
Военно-воздушная академия  
имени профессора Н. Е. Жуковского и Ю. А. Гагарина  
(Россия, г. Воронеж, ул. Старых Большевиков, 54А)  
E-mail: bst76@yandex.ru

**Ф. Х. Аль Сафтли**

адъюнкт,  
Военно-воздушная академия  
имени профессора Н. Е. Жуковского и Ю. А. Гагарина  
(Россия, г. Воронеж, ул. Старых Большевиков, 54А)  
E-mail: fadi79@yandex.ru

**Sergey T. Balanyan**

Candidate of technical sciences, associate professor,  
doctoral candidate of the sub-department  
of aviation armament and effectiveness of combat use,  
Air Force Academy named after professor  
N. E. Zhukovsky and Yu. A. Gagarin  
(54A Sarykh Bol'shevikov street, Voronezh, Russia)

**F. H. Al Saftli**

Adjunkt,  
Air Force Academy named after professor  
N. E. Zhukovsky and Yu. A. Gagarin  
(54A Sarykh Bol'shevikov street, Voronezh, Russia)

**Авторы заявляют об отсутствии конфликта интересов /  
The authors declare no conflicts of interests.**

**Поступила в редакцию/Received 25.08.2021**

**Поступила после рецензирования/Revised 15.10.2021**

**Принята к публикации/Accepted 16.11.2021**