

Научная статья

УДК 004.8+519.6+ 681.5

<http://dx.doi.org/10.17213/1560-3644-2025-3-33-39>

## Метод сверточного градиентного спуска

В.А. Есаулов

Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова,  
г. Новочеркасск, Россия

**Аннотация.** Представлен подход, когда для улучшения качества градиентного спуска градиент обрабатывается скользящим сверточным фильтром. Метод направлен на преодоление ограничений классического градиентного спуска, возникающих в условиях сложной топологии рельефа целевой функции. Предложена нейросетевая интерпретация метода, расширяющая возможности градиентного спуска путем его интеграции в нейросетевые структуры. Проведенные численные эксперименты демонстрируют высокую точность и эффективность метода.

**Ключевые слова:** глобальная оптимизация, целевая функция, градиентный метод, нелинейная функция, нейронная сеть, алгоритм

**Для цитирования:** Есаулов В.А. Метод сверточного градиентного спуска // Изв. вузов. Сев.-Кавк. регион. Техн. науки. 2025. № 3. С. 33–39. DOI: 10.17213/1560-3644-2025-3-33-39.

Original article

## Method of convolutional gradient descent

V.A. Esaulov

Platov South-Russian State Polytechnic University (NPI), Novocherkassk, Russia

**Abstract.** This paper proposes an approach in which the gradient is processed using a sliding convolutional filter to enhance the performance of gradient descent. The method aims to overcome the limitations of classical gradient descent in the context of complex target function topologies. A neural network interpretation of the method is presented, expanding the capabilities of gradient descent through its integration into neural network architectures. Numerical experiments conducted in the study demonstrate the high accuracy and efficiency of the proposed method.

**Keywords:** global optimization, target function, gradient method, nonlinear function, neural network, algorithm

**For citation:** Esaulov V.A. Method of convolutional gradient descent. *Izv. vuzov. Sev.-Kavk. region. Techn. nauki*=*Bulletin of Higher Educational Institutions. North Caucasus Region. Technical Sciences*. 2025;(3):33-39. (In Russ.). DOI: 10.17213/1560-3644-2025-3-33-39.

### Введение

Современные задачи оптимизации в машинном обучении и инженерных дисциплинах требуют эффективных методов работы с нелинейными и высокоразмерными функциями. Традиционные методы оптимизации, такие как градиентный спуск, часто демонстрируют медленную сходимость или застревание в локальных минимумах, особенно для невыпуклых функций потерь [1–6].

Для преодоления этих ограничений активно развиваются адаптивные и гибридные методы, сочетающие классические алгоритмы с нейросетевыми и эволюционными подходами. Такие методы подстраиваются под структуру целевой функции и эффективно используют распределенные вычисления, что особенно важно в условиях высокой вычислительной сложности и ограниченных ресурсов [5–8].

Целью исследования является разработка метода градиентного спуска с использованием скользящих сверточных фильтров для повышения эффективности оптимизации. Научная гипотеза заключается в том, что фильтрация градиента улучшает сходимость и устойчивость алгоритма по сравнению с классическими методами. Работа включает теоретическое обоснование, численный эксперимент и применение метода в задачах машинного обучения.

### Разработка метода сверточного градиентного спуска

Одним из распространенных видов алгоритмов оптимизации являются градиентные методы – это класс алгоритмов оптимизации, которые используют информацию о градиенте (или частных производных) целевой функции для нахождения её минимума (или максимума). Градиентный метод основывается на идее, что для нахождения минимума функции нужно двигаться в направлении, противоположном градиенту, то есть в направлении наибольшего уменьшения функции [1–3].

В данном методе предполагается, что на каждом шаге оптимизации параметры модели обновляются в направлении, противоположном градиенту функции потерь (или целевой функции), с учётом некоторой величины шага (или скорости обучения).

Традиционный метод градиентного спуска имеет вид [1, 2]

$$X_{k+1} = X_k - \lambda_k \nabla F(X_k),$$

где  $X_k, X_{k+1}$  – значения аргумента функции на  $k$ -й и  $(k+1)$ -й итерациях;  $\lambda_k, \lambda_k \geq 0$  – величина шага спуска.

При этом величину шага спуска можно найти из соотношения

$$F(X_k - \lambda_k \nabla F(X_k)) \xrightarrow{\lambda_k} \min.$$

Несмотря на простоту и популярность, градиентные методы имеют ряд недостатков:

1. Медленная сходимость в областях с узкими долинами, где градиент значительно варьируется по направлениям.

2. Чувствительность к шагу обучения: слишком большой шаг может привести к пропуску минимума, а слишком малый замедляет сходимость.

3. Застывание в локальных минимумах или седловых точках, что особенно проблематично для многомерных функций со сложной структурой.

В рамках преодоления указанных недостатков выделим следующие подходы [3–8]:

1. Методы с моментумом (импульсом). Моментум добавляет "инерцию" к предыдущим шагам, что помогает избежать локальных минимумов и ускоряет поиск глобального минимума.

2. Адаптивные методы. Они используют информацию о прошлых шагах для адаптации шага обучения на каждом шаге. Эти методы регулируют величину шага для каждой переменной независимо, что помогает эффективно справляться с функциями с сильно различающимися масштабами в разных направлениях.

3. Использование второго порядка (метод Ньютона и квазиньютоновские методы). Такие методы вычислительно более дорогие, но они значительно ускоряют сходимость за счёт использования информации о гессиане (вторых производных). Это особенно полезно в случае сложных многогранных функций, где стандартный градиентный спуск может не показывать хороших результатов.

Таким образом, приведенные методы имеют как положительные, так и отрицательные особенности. Однако практический опыт численного решения задач оптимизации говорит о том, что точность и скорость сходимости метода во многом определяется характером траектории, проходящей от точки, отвечающей начальному приближению, к конечному значению, достигаемому методом, через промежуточные точки. На качественном уровне можно выделить прямолинейную и зигзагообразную траектории спуска к экстремуму функции.

Траектория, близкая к прямой, может отвечать ситуации выпуклой функции, не имеющей промежуточных локальных минимумов и имеющей гиперповерхности уровня, близкие к сферическим. В случае, когда траектория принимает зигзагообразный характер, сделаем вывод о плохой обусловленности задачи и существенном влиянии рельефа гиперповерхности функции многих переменных на характер поиска её минимума.

Траектория спуска рассматривается как некоторый сигнал, подлежащий обработке и коррекции. В случае, когда сигнал содержит существенно выраженные осцилляции или шумо-

подобные компоненты, для выявления его значимых признаков необходимо применить преобразование фильтрации, которое могло бы сгладить резкие изменения и убрать паразитные составляющие. Поскольку основным параметром, задающим траекторию спуска, является градиент целевой функции, то, комбинируя его компоненты, можно в существенной степени влиять на ход итерационного процесса спуска.

В теории обработки сигналов одним из основных приемов обработки сигналов является фильтрация. При этом фильтр может быть оконным, т.е. преобразовывать ограниченную часть сигнала. Использование оконных фильтров с возможностью их сдвига по области сигнала позволяет выявить локальные частотно-временные характеристики и получить более полную картину структуры [1–4].

Применение оконных фильтров в задачах глобальной оптимизации преследует много целей. Основной является оптимизация траектории спуска к глобальному минимуму функции. Поскольку градиент определяет кратчайшее направление достижения экстремума функции, его адаптивная фильтрация в условиях сложной топологии целевой функции представляется интересным развитием градиентного спуска. В этой ситуации формула градиентного спуска выглядит следующим образом:

$$X_{k+1} = X_k - w_k \cdot \nabla F(X_k), \quad (1)$$

где  $w_k$  – ядро оконного фильтра, выполняющего свертку с градиентом.

Аналогично методу наискорейшего спуска поставим вопрос адаптивного поиска ядра фильтра, чтобы выполнялось соотношение

$$F(X_{k+1}) = F(X_k - w_k \cdot \nabla F(X_k)) \xrightarrow{w_k} \min. \quad (2)$$

Свертка  $w_k \cdot \nabla F(X_k)$  в выражениях (1), (2) имеет несколько интерпретаций. Рассмотрим выражение вида  $F(X_k + w_k \cdot h_k)$ , где  $h_k$  – вектор шага. Разлагая  $F(X_k + w_k \cdot h_k)$  в ряд Маклорена по значениям компонент  $h_k$  с удержанием членов первого порядка, получим следующее представление:

$$\begin{aligned} F(X_k + w_k \cdot h_k) &\approx F(X_k) + h_k^T (w_k \cdot \nabla F(X_k)) = \\ &= F(X_k) + h_k^T (\nabla^{w_k} F(X_k)). \end{aligned} \quad (3)$$

Величина  $\nabla^{w_k} F(X_k) = w_k \cdot \nabla F(X_k)$  трактуется как обобщение градиента на случай вычисления по ядру фильтра  $w_k$ . Это обусловлено тем, что компоненты такого градиента имеют вид

$$\begin{aligned} \left( \nabla^{w_k} F(X_k) \right)_j &= \frac{\partial_{w_k} F(X)}{\partial x_j} \Big|_{X=X_k} = \\ &= \sum_{i=1}^m w_i^k \frac{\partial F(X)}{\partial x_{i+j}} \Big|_{X=X_k}, \quad j = 1 \dots n. \end{aligned}$$

В случае, если величина  $j$  приобретает значение, которое приводит к выходу индексов за границы размерности градиента, во избежание некорректности введем дополнительные граничные условия, доопределяющие компоненты градиента при выходе за его границы, например, нулевые или периодические граничные условия. Если рассматривать градиент как вектор частных производных, то компоненты (3) интерпретируются как частные сверточные производные  $\frac{\partial_{w_k} F(X)}{\partial x_j}$ , вычисленные от функции  $F(X)$  по ядру фильтра  $w_k$ , определенные в точке  $X = X_k$ . В этом отношении сверточная производная является более общим понятием по сравнению с производной по направлению.

С другой стороны, операция свертки также  $w_k \cdot X$  трактуется как произведение циркулянтной матрицы  $C_{w_k}$ , порожденной сдвигами  $n$ -мерного вектора, образованного  $w_k$  и дополненного нулями до вектора длины  $n$ . В этом отношении  $w_k \cdot X$  рассматривается как отображение  $W \cdot \hat{X} \rightarrow \hat{X}$ , где  $W$  является пространством сверточных операторов ( $\hat{X}$  является левым модулем над  $W$ ). Таким образом, формула (1) получена путем преобразования линейного векторного пространства в модуль над кольцом сверточных операторов, вычислительная схема приведена на рис. 1.

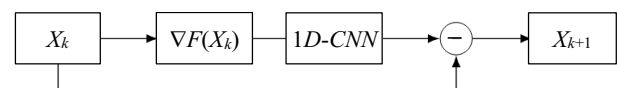


Рис. 1. Вычислительная схема сверточного градиентного спуска

Fig. 1. Computational scheme of convolutional gradient descent

В вычислительной схеме блок 1D-CNN выполняет свертку  $w_k \cdot \nabla F(X_k)$ . В терминах вычислительной схемы процесс (1) интерпретируется как элементарная сверточная нейронная сеть. Векторный вход  $X_k$  подается на блок вычисления градиента функции. Выход блока вычисления градиента, в свою очередь, является

входом для одномерного сверточного слоя  $1D-CNN$ . Далее в блоке вычитания происходит вычисление разности  $X_k - w_k \cdot \nabla F(X_k)$ , которая рассматривается как выходной слой нейросети. Функцией потерь нейросети, обладающей архитектурой, является соотношение (2) (см. рис. 1).

Интерпретация соотношений (1), (2) в терминах нейросетевой концепции интересна тем, что задача нелинейной оптимизации переформулирована как задача обучения некоторой нейросети. К очевидным преимуществам можно отнести понижение размерности, заключающееся в том, что размерность ядра  $w_k$  существенно меньше размерности  $X_k$ . Операция свертки  $w_k \cdot \nabla F(X_k)$ , по сравнению с операцией умножения  $\nabla F(X_k)$  на скалярный шаг спуска, дает более широкие возможности для выполнения поиска на многомерном рельефе функции  $F(X)$ . Это обусловлено тем, что, помимо масштабирования вектора градиента по его длине, свертка выполняет преобразование поворота, что важно при исследовании функции со сложной топологией. Ввиду наличия звена вычитания между входом и модифицированным градиентом, нейросеть, реализованная на базе схемы (см. рис. 1), классифицирована как имеющая элементы *ResNet*-архитектуры.

Вместе с тем нейросетевая концепция оптимизации функции многих переменных имеет свои ограничения. Известным фактом в области нейросетевых вычислений является то, что для корректной работы нейронной сети с данными зачастую необходима их нормализация. Также в функционировании нейронных сетей большую роль играют функции активации, которые ограничивают динамический диапазон выходных данных. Все это требует некоторой адаптации задачи минимизации функции  $F(X)$ .

Без ограничения общности введем гиперкуб  $\Omega = [a, b]^n \subset R^n$ , такой, что задача поиска минимума  $F(X)$  будет решаться для  $X \in \Omega$ . Тогда задачу минимизации  $F(X)$  перепишем в виде

$$f(\xi) = F(a + (b - a) \cdot \xi) \xrightarrow{\xi} \min, \quad \xi \in [0, 1]^n. \quad (4)$$

В постановке (4) задача минимизации на гиперкубе сведена к задаче поиска нормализованного вектора  $\xi \in [0, 1]^n$ , доставляющего минимум целевой функции. Такое масштабирование дает возможность использовать его в качестве входных данных.

Итерационный процесс (1) рассматривается как вычисление выхода нейросети, в котором функцией активации является единичное преобразование. В случае введения функции активации  $\phi$ , поэлементно применяемой к элементам выхода, итерационный процесс с учетом введенной нормализации (4) приобретает вид

$$\xi_{k+1} = \phi(\xi_k - w_k \cdot \nabla f(\xi_k)), \quad (5)$$

где (5) интерпретируем как выполнение нелинейного проецирования вектора  $\xi_k - w_k \cdot \nabla f(\xi_k)$  на некоторое множество, где  $f(\xi_{k+1}) \xrightarrow{w_k} \min$ .

При определенном виде функции активации соотношение (5) рассмотрим как проксимальный сверточный градиентный алгоритм. Так, если  $\phi$  задается как *ReLU*-функция, то (5) интерпретируется как аналог проксимального шага для  $l_1$ -регуляризации [6, 7]. Ограниченный диапазон функции активации, с учетом введенной нормализации, способствует повышению устойчивости и гибкости вычислительного процесса минимизации  $F(X)$ .

Нейронную сеть, реализованную на базе схемы (см. рис. 1), снабженную выходной функцией активации  $\phi$ , рассмотрим как элементарный вычислительный блок сверточного градиентного спуска, который лежит в основе более сложных решений по поиску минимума нелинейной функции.

Выстраивая ансамбли из параллельных блоков градиентного спуска, отличающихся характером и размерностью ядра, выполним многомасштабную коррекцию градиента. Фильтры меньшей длины в составе слоя  $1D-CNN$  блока спуска определяют локальный характер поведения градиента. Увеличение длины фильтра позволяет формировать более глобальное направление поиска минимума функции. Таким образом, ансамблирование блоков позволяет получать результаты, сочетающие в себе как локальный, так и глобальный поиск. Выходы элементов ансамбля можно усреднять, получая результат, который носит как итоговый, так и промежуточный характер. Интерес также представляетconcatenation выходов в некоторый вектор большей размерности. К нему применяется какое-либо нелинейное преобразование для формирования выхода или промежуточного представления.

Последовательное объединение блоков градиентного спуска задает многошаговый градиентный спуск, где каждый блок обрабатывает выход предыдущего блока в контексте соответствующей ему структуры направления поиска. Таким образом, последовательно-параллельное соединение блоков спуска сочетает возможности пространственно-временного исследования данных с целью поиска минимума функции.

Применение нейросети в решении задачи минимизации предполагает, что в качестве обучающей выборки используется не одно, а некоторое количество начальных приближений, которые в нормализованном варианте распределены в единичном гиперкубе. Так, если выходные результаты, полученные в процессе обучения нейросети не обеспечивают желаемого уровня функции потерь, нейросеть дополнительно обучается, используя в качестве обучающей выборки уже их. Итеративный характер метода градиентного спуска также создает предпосылки к тому, что решение задачи нелинейной оптимизации функции носит многоэтапный характер. Таким образом, дообучение нейросети, на результатах предыдущего цикла обучения может способствовать уточнению значений точки, в которой достигается минимум функции.

### Численный эксперимент

Для проверки адекватности предложенной концепции нейросетевой реализации поиска минимума функции на базе соотношений (1), (5) проведен численный эксперимент. В качестве тестовых функций выбаны функции Розенброка, Леви и Экли [4–6]. Они отличаются достаточно сложным рельефом и в точке минимума принимают нулевое значение.

Численный эксперимент состоял в выполнении  $k$  прогонов минимизации тестовой функции при различных начальных приближениях. Вычисления выполнены в облачном сервисе *Kaggle* с использованием четырехпроцессорной конфигурации с архитектурой  $\times 86\_64$ , 31 Гб ОЗУ, жестким диском объемом 29 Гб, а также при поддержке ускорителя *GPU NVIDIA Tesla P100* с объемом видеопамяти 16 Гб.

При подготовке процесса минимизации функции в рамках прогона выполнены следующие подготовительные действия:

1. В качестве обучающей выборки  $X$  и совокупности начальных приближений сгенерировано множество из  $N$  точек, равномерно распределенных в гиперкубе  $[0,1]^n$ .

2. Создана нейросеть с функцией потерь, в качестве которой указана тестовая функция в форме (4) и размером минибатча  $l$ .

Процесс минимизации представляет собой циклический процесс, выполняющийся за  $d$  повторений, содержащий следующие действия:

1. Обучение нейросети на основе обучающей выборки  $X$  за  $m$  эпох.

2. Вычисление выхода нейросети  $Y$  при подаче на вход данных  $X$ .

3. Замена выборки  $X$  на данные выхода  $Y$ .

По окончании процесса минимизации сформированы промежуточные результаты, после чего проведен следующий прогон.

Значения основных гиперпараметров приведены в табл. 1.

Таблица 1

Table 1

**Основные гиперпараметры численного эксперимента**  
**Main hyperparameters of the numerical experiment**

Гиперпараметр эксперимента	Значение параметра
Число прогонов процесса минимизации $k$	100
Размерность пространства $n$	40
Число циклов обучения $d$	10
Число эпох обучения на цикл $m$	100
Число элементов в обучающей выборке $N$	1000
Размер минибатча $l$	32
Интервал гиперкуба $[a, b]$ , в котором проводился поиск минимума функции	$[-2,2]$

Для поиска минимума тестовой функции использована нейросетевая архитектура, имеющая следующую структуру:

1. Входом нейросети является вектор длины  $n$ .

2. Данные входа подаются в параллельно соединенные блоки сверточного градиентного спуска с размерами сверточных фильтров 2 и 4.

3. Выходы блоков сверточного градиентного спуска усредняются в слое усреднения.

4. Выход слоя усреднения является входом для сигмоидальной функции активации, вычисляемой для каждого элемента входа.

5. Выход функции активации подается на вход блока сверточного градиентного спуска с размером сверточного фильтра, равным 8.

Основные результаты минимизации тестовых функций представлены в табл. 2.

Таблица 2

Table 2

Результаты минимизации тестовых функций  
Results of minimization of test functions

Основные параметры решения	Функция Леви	Функция Экли	Функция Розенброка
Минимальное значение	$8,19e-06$	$3,28e-05$	$4,26e-10$
Максимальное значение	$1,1e-05$	$6,62e-05$	$0,0001$
Среднее значение	$1,02e-05$	$3,61e-05$	$2,11e-05$
Стандартное отклонение	$8,03e-07$	$4,62e-06$	$2,55e-05$
Среднее число итераций	915	761	937

Статистика времени выполнения прогонов минимизации тестовых функций приведена в табл. 3.

Таблица 3

Table 3

Время выполнения минимизации тестовых функций  
Results of minimization of test functions

Время минимизации	Функция Леви	Функция Экли	Функция Розенброка
Минимальное значение, с	70,7	70,1	70,6
Максимальное значение, с	80,8	75,9	74,6
Среднее значение, с	72,01	71,2	71,9
Стандартное отклонение, с	1,57	1,001	0,81

На рис. 2 представлены графики усредненных по количеству циклов обучения зависимостей изменения значений функции потерь Леви, Экли и Розенброка от числа эпох с доверительным интервалом 95 %.

С целью повышения наглядности для представления значений функций потерь выбран логарифмический масштаб.

Результаты численного эксперимента (см. табл. 2, рис. 2), подтверждают высокую точность метода. Как отмечено выше, точное значение тестовых функций в точке минимума равно нулю. Таким образом, приближенные значения минимума данных функций интерпретированы как абсолютная погрешность, диапазон которой составляет  $10^{-10} - 10^{-4}$  (см. табл. 2). Диапазон найденных минимумов функций Леви и Экли демонстрирует низкое стандартное отклонение, что указывает на стабильность. Для функции Розенброка наблюдается большая вариативность, вероятно, из-за сложной структуры с узкими долинами.

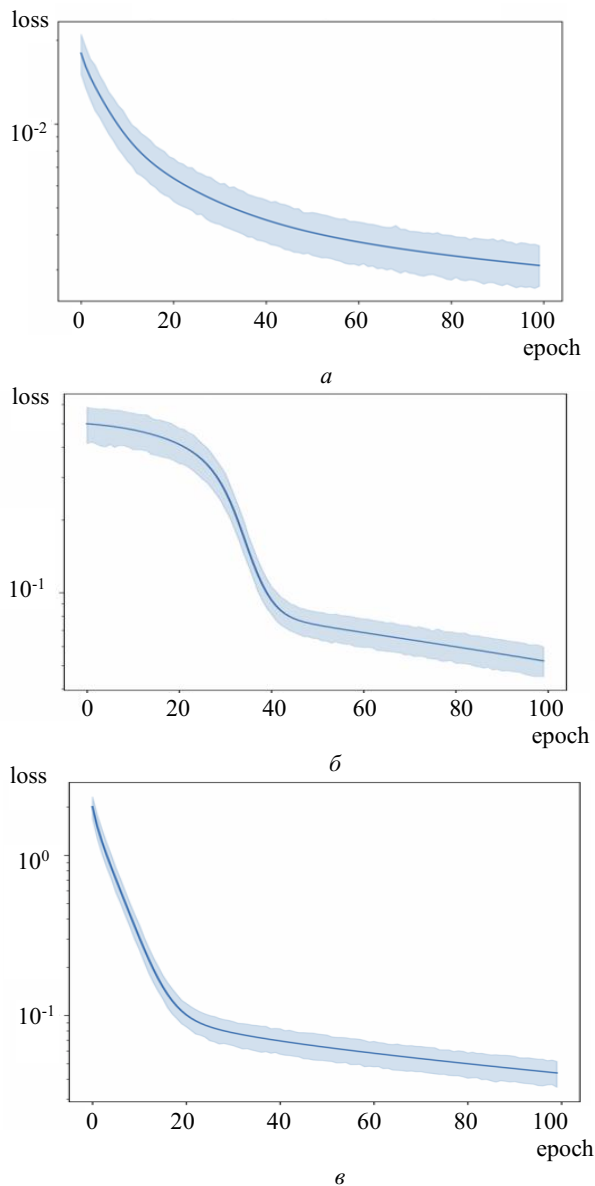


Рис. 2. Графики изменения значений функции

Леви (а), Экли (б), Розенброка (в)

Fig. 2. Graph of changes in the values functions of the

Levi (a) Ackley (b), Rosenbrock (c)

Среднее число итераций остается приемлемым для практических задач оптимизации, несмотря на высокую нелинейность функций и размерность пространства. Уровень значений границ доверительных интервалов значений тестовых функций свидетельствуют о стабильности диапазона изменения результатов в условиях задания разных начальных приближений.

Из данных о времени выполнения алгоритма (см. табл. 3) сделан вывод о близости их значений. Характер распределения стандартного отклонения времени прогона алгоритма свидетельствует о том, что его уровень достаточно высок в условиях поиска минимума многоэкстремальных функций, к которым относятся функции Леви и Экли.



## Заключение

Предложенный метод сверточного градиентного спуска (1), (2) расширяет возможности традиционного градиентного спуска за счет применения скользящих сверточных фильтров, это улучшает управление направлением спуска. Метод интерпретируется как нейронная вычислительная схема и позволяет использовать нейросетевые решения для решения задач нелинейной оптимизации.

При проведении вычислительного эксперимента использована нейросетевая архитектура, содержащая комбинацию различных соединений блоков сверточного спуска. Его результаты позволили сделать вывод о стабильности процесса поиска минимумов тестовых функций и высоком уровне точности полученных целевых значений, что свидетельствует об адекватности предложенного метода.

## Список источников

1. Beck A. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics. 2017. DOI: 10.1137/1.9781611974997.

2. Nesterov Y. Introductory Lectures on Convex Optimization. Springer US. 2004. DOI: 10.1007/978-1-4419-8853-9.
3. Yue P., Fang C., Lin Z. On the lower bound of minimizing polyak-lojasiewicz functions // The Thirty Sixth Annual Conference on Learning Theory. Proceedings of Machine Learning Research. 2023. 195. 2948–2968. <https://proceedings.mlr.press/v195/yue23a/yue23a.pdf>
4. Loizou N., Richtárik P. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. Comput Optim Appl 2020;(77):653-710. DOI: 10.1007/s10589-020-00220-z.
5. Gao S, Pei Z, Zhang Y, Li T. Bearing fault diagnosis based on adaptive convolutional neural network with Nesterov momentum. IEEE Sens J 2021;21(7): 9268-9276. DOI: 10.1109/JSEN.2021.3050461.
6. Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: учеб. для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 440 с.
7. Nikolovski F. et al. Gradient Descent Methods for Regularized Optimization //arXiv preprint arXiv: 2412.20115. 2024.
8. Безгачев Ф.В. Методы глобальной оптимизации на основе различных подходов оптимизации // E-Scio. 2020. №12 (51). С. 470–476.

## References

1. Beck A. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics. 2017. DOI: 10.1137/1.9781611974997.
2. Nesterov Y. Introductory Lectures on Convex Optimization. Springer US. 2004. DOI: 10.1007/978-1-4419-8853-9.
3. Yue P., Fan C., Lin Z. On the Lower Bound of Minimizing Polyak-Lojasiewicz Functions. The Thirty-Sixth Annual Conference on Learning Theory. *Proceedings of Machine Learning Research*. 2023;(195):2948-2968.
4. Loizou N., Richtárik P. Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point, and Subspace Descent Methods. *Computational Optimization and Applications*. 2020;(77):653-710. DOI: 10.1007/s10589-020-00220-z.
5. Gao S., Pei Z., Zhang Y., Li T. Bearing Fault Diagnosis Based on Adaptive Convolutional Neural Network with Nesterov Momentum. *IEEE Sensors Journal*. 2021;21(7):9268–9276. DOI: 10.1109/JSEN.2021.3050461
6. Attetkov A. V., Galkin S. V., Zarubin V. S. *Optimization Methods: Textbook for Universities*. Moscow: Publishing House of Bauman Moscow State Technical University; 2001. 440 p.
7. Nikolovski F. et al. *Gradient Descent Methods for Regularized Optimization*. 2024. arXiv preprint arXiv:2412.20115.
8. Bezgachev F.V. Global Optimization Methods Based on Various Optimization Approaches. *E-Scio*. 2020;12(51):470-476.

## Сведения об авторах

Есаулов Василий Александрович – канд. техн. наук, доцент, кафедра «Программное обеспечение вычислительной техники», esaul\_va@mail.ru

## Information about the authors

Vasily A. Esaulov – Cand. Sci. (Eng.), Associate Professor, Department «Software for Computer Engineering», esaul\_va@mail.ru

**Статья поступила в редакцию / the article was submitted 23.05.2025;**  
**одобрена после рецензирования / approved after reviewing 10.06.2025;**  
**принята к публикации / accepted for publication 16.06.2025.**