

УДК 004.8

DOI: 10.25559/SITITO.16.202002.351-357

Оптимизация гиперпараметров сверточной нейронной сети, используемой при построении двумерных карт местности

А. В. Акинина, М. Б. Никифоров*

ФГБОУ «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», г. Рязань, Россия

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1

*nikiforov.m.b@mail.ru

Аннотация

Рассматривается задача определения гиперпараметров нейронной сети в контексте построения двумерной карты местности. Целью работы является анализ существующих методов оптимизации гиперпараметров, а также разработка улучшенного подхода к определению гиперпараметров на основе достоинств и недостатков существующих методов. В качестве гиперпараметров рассматриваются скорость обучения, коэффициент регуляризации, размер мини-батча, вероятность дропаута, сдвиг и растяжение, применяемые в пакетной нормализации. Среди существующих методов подбора гиперпараметров были рассмотрены случайный поиск, поиск по сетке (вариация параметров), Байесовская оптимизация, эволюционная оптимизация, оптимизация на основе градиентов и спектральный метод. На основе исследуемых методов был предложен подход, который увеличивает производительность и качество работы алгоритма. Целевая функция оптимизируется в области, описываемой простыми ограничениями на нижнюю и верхнюю границу, гиперпрямоугольник, а переменные решения ограничены целочисленными значениями. Алгоритм основывается на исходной модели целевой функции, а затем выбирает новые точки для оценки, пытаясь сбалансировать исследование неизвестных значений и уже найденных. Одноцелевая задача оптимизации для выбора новой точки решается простым генетическим алгоритмом. Критерием остановки является максимально допустимое число оценок или установленный временной порог. Оценка целевой функции реализуется с помощью очереди задач. Применение метода оптимизации без производных позволяет сократить вычислительные потери за счет оптимизации целевой функции в области, описываемой простыми ограничениями на нижнюю и верхнюю границу. При этом точность алгоритма оказывается выше, чем при использовании случайного поиска, байесовской и спектральной оптимизации.

Ключевые слова: сверточная нейронная сеть, поиск по сетке, случайный поиск, эволюционная оптимизация, Байесовская оптимизация, оптимизация на основе градиентов, спектральный метод.

Для цитирования: Акинина, А. В. Оптимизация гиперпараметров сверточной нейронной сети, используемой при построении двумерных карт местности / А. В. Акинина, М. Б. Никифоров. – DOI 10.25559/SITITO.16.202002.351-357 // Современные информационные технологии и ИТ-образование. – 2020. – Т. 16, № 2. – С. 351-357.

© Акинина А. В., Никифоров М. Б., 2020



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Hyperparameter Optimization of CNN for Map Building

A. V. Akinina, M. B. Nikiforov*

Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russia
59/1 Gagarin St., Ryazan 390005, Russia

*nikiforov.m.b@mail.ru

Abstract

This article describes an approach for solving the task of finding hyperparameters of an artificial neural network, which is used for making a 2D land map. The main goal of research was an analysis of methods for finding hyperparameters and creating a better method for solving this task, which would be based on existing methods. We considered on various hyperparameters such as velocity of training, coefficient of regularization, size of batch, probability of drop out, shifting, used for batch normalization. Among existing methods for finding hyperparameters we considered on the random search method, searching by grid, the Bayesian optimization, the evolution algorithm, the optimization, based on gradients, and the spectral method. As a result, we created a new method for finding hyperparameters which showed a better result in most of the use cases, which we have (mostly for middle European part of Russia).

The main idea of the method for finding hyperparameters is consisted in an approach for optimization of the quality function with a simple condition for lower and upper limits and a demand that the value of the function needed to be an integer number. This task may be solved with a simple genetic algorithm. Using the optimization algorithm without evaluating derivatives gives decreasing time complexity of the algorithm without losing quality of the algorithm. In many cases the quality of result was better than results of existing methods.

Keywords: Convolutional neural network, grid search, random search, evolutionary optimization, Bayesian optimization, gradient based optimization, spectral method.

For citation: Akinina A.V., Nikiforov M.B. Hyperparameter Optimization of CNN for Map Building. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2020; 16(2):351-357. DOI: <https://doi.org/10.25559/SITITO.16.202002.351-357>



Введение

При построении карт местности с использованием нейронных сетей необходимо решить задачу выбора наилучшего набора параметров для моделей машинного обучения, то есть такого набора параметров, который дает наилучшую производительность на доступном наборе данных. При большом объеме данных обучение и тестирование кортежа гиперпараметров может занять много времени — несколько часов и более. Более того, выбор значений гиперпараметров, которые дают высокую точность, иногда является сложным процессом, который обычно начинается с подбора по эвристическим правилам, а затем гиперпараметры подстраиваются вручную. Существует множество алгоритмов определения гиперпараметров для нейронной сети. Для определения гиперпараметров при решении задачи построения карт местности с помощью нейронной сети лучше всего подходит алгоритм математической оптимизации с ограничениями, целью которой является максимизация точности прогнозирования на тестовом наборе данных¹.

Теоретическая часть

К гиперпараметрам относятся:

η — скорость обучения (англ. learning rate);

λ — коэффициент регуляризации;

N_b — размер мини-батча;

p — вероятность дропаута;

γ и β — сдвиг и растяжение, применяемые в пакетной нормализации (англ. BatchNormalization).

В рассматриваемой задаче количество слоев не относится к определяемым гиперпараметрам, поскольку архитектура сети была определена заранее.

Для обучения нейронной сети используется измененный метод градиентного спуска. Проход производится не по всему обучающему множеству (как в пакетном градиентном спуске (англ. Batch GD)), а по мини-батчам (англ. Mini-batch). Таким образом, первоначально сеть обучается всего на нескольких примерах, количество которых и определяется параметром N_b .

Смысл R2-регуляризации заключается в упрощении модели (значения весов становятся меньше) путем добавления к функции потерь дополнительного слагаемого

$$\frac{\lambda}{2n} \|w\|^2, \lambda > 0.$$

Применение дропаута (англ. dropout) необходимо для улучшения эффективности обучения и качества результата. В процессе обучения в выбранных слоях с вероятностью p (параметр алгоритма) выбираются нейроны, которые выключаются из дальнейших вычислений, а более обученные нейроны получают в сети больший вес. Результат дропаута выглядит как измененная функция активации:

$$\alpha(f) = D \otimes f(z),$$

где $D = (X_1, \dots, X_{d_z})$ — d_z -мерный вектор случайных величин, распределенных по закону Бернулли.

Применение дропаута к i -му нейрону выглядит следующим образом:

$$X_i f\left(\sum_{k=1}^{d_i} w_k x_k + b\right) = \begin{cases} f\left(\sum_{k=1}^{d_i} w_k x_k + b\right), & X_i = 1, \\ 0, & X_i = 0. \end{cases}$$

На этапе обучения нейрон остается в сети с вероятностью $q = 1 - p$.

На этапе тестирования необходимо эмулировать поведение ансамбля нейронных сетей, использованного при обучении, для этого функция активации умножается на коэффициент q .

$$X_i f\left(\sum_{k=1}^{d_i} w_k x_k + b\right) \text{ — на этапе обучения.}$$

$$q f\left(\sum_{k=1}^{d_i} w_k x_k + b\right) \text{ — на этапе тестирования.}$$

Удобнее использовать обратный дропаут, поскольку в этом случае модель описывается лишь однажды, а потом запускается обучение и тестирование на этой модели, меняется только параметр p . В случае прямого дропаута необходимо изменять нейронную сеть для проведения тестирования, поскольку без умножения на q нейрон возвращает значения выше, чем те, которые ожидают получить последующие нейроны.

В случае обратного дропаута функция активации умножается на коэффициент (равен обратной величине вероятности того, что нейрон остается в сети

$$\frac{1}{1-p} = \frac{1}{q} \text{ не во время теста, а во время обучения.}$$

$$\frac{1}{q} X_i f\left(\sum_{k=1}^{d_i} w_k x_k + b\right) \text{ — на этапе обучения.}$$

$$f\left(\sum_{k=1}^{d_i} w_k x_k + b\right) \text{ — на этапе тестирования.}$$

Слой из n нейронов на отдельном шаге этапа обучения можно рассматривать как ансамбль из n экспериментов Бернулли с вероятностью успеха p . Так как каждый нейрон представлен в виде независимой случайной величины, распределенной по закону Бернулли, то общее число исключенных нейронов — случайная величина, имеющая биномиальное распределение. При обучении нейронной сети используется пакетная нормализация (англ. batch normalization, BN) — нормализация каждого входа отдельно, по определенному пакету, а не по всему набору данных. Для очередного пакета $P = \{x_1, \dots, x_m\}$ необходимо:

1) вычислить математическое ожидание и дисперсию:

$$\mu_p = \frac{1}{m} \sum_{i=1}^m x_i, \sigma_p^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_p)^2;$$

¹ Акинин, М. В. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / М. В. Акинин, М. Б. Никифоров, А. И. Таганов. – М.: Горячая линия – Телеком, 2015; Акинин, М. В. Интеллектуальные алгоритмы построения двумерных карт местности / М. В. Акинин, Н. В. Акинина, М. Б. Никифоров. – М.: КУРС, 2020.



2) нормализовать входы:

$$x_i = \frac{x_i - \mu_p}{\sqrt{\sigma_p^2 + \varepsilon}};$$

3) вычислить результат:

$$y_i = \gamma x_i + \beta,$$

где γ и β — сдвиг и растяжение, обучаются тоже градиентным спуском, как и веса.

Задача оптимизации гиперпараметров состоит в нахождении значений параметров алгоритма машинного обучения для достижения максимальной точности прогнозирования в наборе данных. Гиперпараметры нужно подобрать так, чтобы модель за приемлемое время оптимально решила задачу обучения, а именно разбила входное изображение на классы объектов, которые в дальнейшем будут нанесены на карту местности. Кортеж гиперпараметров дает оптимальную модель, которая оптимизирует заданную функцию потерь при заданных данных. Сеть прогоняется с различными комбинациями гиперпараметров, а затем решение принимается на основе их производительности на валидационном множестве.

Подбор гиперпараметров может осуществляться следующими способами:

1. случайный поиск;
2. поиск по сетке (вариация параметров) – полный перебор по заданному вручную подмножеству пространства гиперпараметров, легко параллелизуем, поскольку параметры не зависят друг от друга, сопровождается измерением производительности (перекрестная проверка на тренировочном множестве) или прогонкой алгоритма на устоявшемся проверочном наборе;
3. Байесовская оптимизация;
4. эволюционная оптимизация;
5. оптимизация на основе градиентов — для конкретных алгоритмов обучения определяется градиент гиперпараметров, оптимальное значение которых устанавливается с помощью градиентного спуска;
6. спектральный метод.

Суть байесовской оптимизации заключается в построении стохастической модели функции отображения из значений гиперпараметров в целевую функцию, примененную на множестве проверки. Алгоритм является итеративным, применяются перспективные значения гиперпараметров, а затем обновляются. С каждым шагом накапливается как можно больше информации о целевой функции, о ее оптимальном значении. Важной особенностью этого метода является баланс между поиском новых значений гиперпараметров и использованием тех параметров, значения которых близки к оптимальным.

Алгоритм эволюционной оптимизации (улучшенный генетический алгоритм) состоит в следующем.

Случайным образом генерируется популяция, состоящая из 100 кортежей.

Производится оценка кортежей, с помощью перекрестной проверки десятикратной точности алгоритма рассчитывается функция приспособленности.

Кортежи гиперпараметров ранжируются по их относительной пригодности.

Кортежи гиперпараметров с худшей производительностью за-

меняются на новые кортежи гиперпараметров, образованные скрещиванием и мутацией.

Повторяются шаги 2—4, пока не достигается удовлетворительная производительность алгоритма или пока производительность не перестанет улучшаться.

Метод спектральной оптимизации основан на эффективных алгоритмах для обучения деревьев решений с использованием дискретного преобразования Фурье. На выходе получаются границы сложности выборки для обучения деревьев решений при сопоставлении современных границ времени выполнения (полиномиальный и квазиполиномиальный, соответственно). Методология основана на решении задачи математической оптимизации с ограничением и применении алгоритма оптимизации без производных для такой задачи. Алгоритм оптимизации без производных — это алгоритм, который не требует информации о производных любого порядка.

Если скорость обучения слишком мала, то даже после обучения нейронной сети в течение длительного времени она будет далека от оптимальных результатов. С другой стороны, если скорость обучения слишком высока, то сеть очень быстро выдаст ответы.

Было принято решение использовать изменяющуюся скорость обучения.

$$\eta = \frac{0.001}{1 + 0.005 * \text{stepcount}}.$$

Большой начальный шаг обучения (0.001) позволяет сети быстрее сходиться к минимуму в начале обучения, а малый шаг на поздних стадиях позволяет ближе подойти к оптимальному состоянию сети.

Пусть X описывает множество всех допустимых назначений значений для всех гиперпараметров. Например, в простейшем случае, когда все гиперпараметры принимают неограниченные действительные значения, множество X можно интерпретировать как вещественное n -мерное векторное пространство (где n — число гиперпараметров). Сложность заключается в том, что каждый гиперпараметр имеет свои ограничения, которые необходимо учитывать.

В качестве метрики производительности будет выступать точность – доля правильно классифицированных объектов, поскольку эта метрика наиболее удобна в задачах классификации.

Задача определения наилучших гиперпараметров может быть сформулирована следующим образом:

$$\arg \max_x \{f(x, D) : x \in X\},$$

где f – целевая функция, т.е. оценка производительности нейронной сети, D – набор данных, x – конкретный кортеж гиперпараметров.

Поскольку функцию f не всегда можно представить аналитически, использование классических методов численной оптимизации невозможно.

Для вычисления f в одной точке может потребоваться достаточно много времени, которое включает в себя время на обучение модели прогнозирования на большом наборе данных и тестирование ее производительности на валидационном множестве с использованием кросс-энтропии. Таким образом, оценка целевой функции является вычислительно дорогой. Поэтому в данном случае не подходят генетические алгорит-



мы, в которых предусмотрено вычисление большего количество оценок целевой функции.

Для решения данной задачи подойдет метод метрической стохастической поверхности отклика [13] с небольшими изменениями:

$$\arg \max \{f(x) : x \in [E^L, x^U] \subset R^n\},$$

где E^L, x^U — нижняя и верхняя границы решений, x — целые числа.

Целевая функция оптимизируется в области, описываемой простыми ограничениями на нижнюю и верхнюю границу, гиперпрямоугольником, а переменные решения ограничены целочисленными значениями. Таким образом, данная оптимизация представляет собой черный ящик с ограничениями.

Алгоритм основывается на исходной модели целевой функции f , а затем выбирает новые точки для оценки, пытаясь сбалансировать исследование неизвестных значений и уже найденных. Исходная модель необходима для определения потенциального глобального оптимума. В качестве такой модели выступает модель радиальной базисной функции в сочетании с полиномиальным хвостом степени 1.

Алгоритм начинается с оценки целевой функции в $n+1$ точках, выбранных случайным образом. S — множество точек, в которых целевая функция была оценена. Затем на каждой итерации алгоритм подбирает модель для f , которая интерполирует точки в S и выбирает следующую точку оценки (y) в соответствии с критериями, которые должны быть максимизированы. Первый критерий — значение в точке y . Второй — евклидово расстояние от y до ближайшей точки в S .

Поскольку необходимо определить одну точку y , задача трансформируется в однокритериальную. Оба критерия нормализуются и сводятся к одному путем рассмотрения взвешенной комбинации с весом w , который определяет компромисс. Вес w выбирается в соответствии с циклической стратегией, которая выбирает между критерием \max - \min расстояния для исследования неизвестных значений и критерием выбора точек, которые должны иметь большое значение для целевой функции для задачи максимизации согласно используемой модели. Результирующая одноцелевая задача оптимизации для выбора y решается простым генетическим алгоритмом. Генетический алгоритм применяется только к вспомогательной задаче оптимизации,

как только значение y определено, f оценивается в y , точка добавляется к S , и итерация завершена. Алгоритм выполняет итерацию по этой схеме до тех пор, пока не будет удовлетворен критерий остановки, обычно основанный на максимально допустимом числе оценок f или на максимальном времени центрального процессора.

Оценка целевой функции является трудоемкой операцией, поэтому эту задачу необходимо распараллелить. Для этого используется очередь задач. Эти задачи могут быть двух видов. Первый — это оценка целевой функции в новой точке, а второй — это вычисление новой точки поиска, в которой должна оцениваться целевая функция. Пока имеются доступные процессоры, задача удаляется из очереди и назначается процессору. Из-за более длительного вычислительного времени задачи вида 1 всегда имеют приоритет над задачами вида 2. В задачах одного вида применяется правило «первым пришел — первым обслужен».

Когда задача вида 1 завершается, она выдает новую точку интерполяции, которая добавляется к набору S . Когда задача вида 2 завершается, происходит проверка, должна ли новая определенная точка поиска отбрасываться из-за нескольких используемых критериев. Если точка поиска принята, то необходимо решить задачу вида 1, чтобы оценить значение целевой функции в ней.

Может произойти такая ситуация, когда при выполнении оценки в точке y , та же самая точка y генерируется как точка поиска параллельными задачами вида 2, и, следовательно, оценка $f(y)$ выполняется несколько раз.

Чтобы избежать такой ситуации необходимо выполнить следующее. Когда задача вида 1 передается на обработку, к S добавляется временный узел интерполяции в точке y со значением целевой функции, определяемым значением существующей модели в точке y . Поскольку следующая точка поиска никогда не может совпадать с узлом интерполяции, даже в последовательной версии алгоритма (требуется минимальное расстояние от существующих узлов интерполяции), это гарантирует, что новые точки поиска будут отличаться. Временный узел интерполяции удаляется, как только соответствующая задача вида 1 завершена.

Некоторые решения, принимаемые алгоритмом оптимизации, зависят от разницы между наибольшим и наименьшим значением функции среди узлов интерполяции. Добавление временного узла интерполяции, расширяющего диапазон известных значений функций, может изменить эти решения, и это особенно рискованно, если это результат неточной суррогатной модели с большими колебаниями. Поэтому временные узлы интерполяции не могут расширять диапазон между известными значениями функций. Это достигается путем ограничения значения модели временного узла интерполяции до диапазона существующих значений функции.

Экспериментальные исследования

По результатам экспериментов байесовская оптимизация показала лучшие результаты с меньшими вычислениями по сравнению с поиском по решётке и случайным поиском, поскольку можно сделать вывод о качестве экспериментов до их выполнения. Результаты экспериментов приведены в таблице 1, а именно данные о точности набора тестов для наилучшей конфигурации гиперпараметров, полученной с помощью того или иного алгоритма. По итогам экспериментов поиск по сетке и эволюционный алгоритм показали плохие временные результаты, поэтому в итоговую таблицу по сравнению точности не вошли. Используется одно и то же пространство гиперпараметров для всех методологий. На заключительном этапе каждый прогон каждого алгоритма позволяет исследовать ровно 100 различных конфигураций гиперпараметров без ограничения по времени.

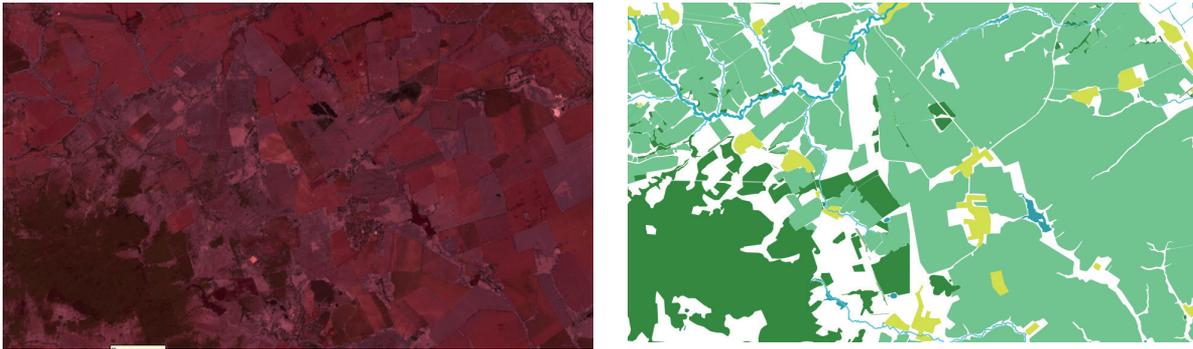
Исходный спутниковый снимок и результат представлены на рис. 1.

Таблица 1. Результаты эксперимента

Table 1. Experiment results

Алгоритм	Точность
Байесовская оптимизация	97,51
Спектральная оптимизация	98,34
Случайный поиск	89,02
Оптимизация без производных	97,96





Р и с. 1. Исходный спутниковый снимок и результат работы алгоритма
F i g. 1. The original satellite image and the result of the algorithm

Заключение

В статье рассмотрены методы подбора гиперпараметров для модели нейронной сети при построении двумерных карт местности. На основе исследуемых методов был предложен подход, который увеличивает производительность и качество работы алгоритма. Применения метода оптимизации без производных позволяет сократить вычислительные потери за счет оптимизации целевой функции в области, описываемой простыми ограничениями на нижнюю и верхнюю границу. При этом точность алгоритма оказывается выше, чем при использовании случайного поиска, байесовской и спектральной оптимизации.

References

- [1] Snoek J., Larochelle H., Adams R.P. Practical Bayesian optimization of machine learning algorithms. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*. Vol. 2 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA; 2012. p. 2951-2959. (In Eng.)
- [2] Hutter F., Hoos H.H., Leyton-Brown K. Sequential Model-Based Optimization for General Algorithm Configuration. In: Coello C.A.C. (ed.) *Learning and Intelligent Optimization*. LION 2011. *Lecture Notes in Computer Science*. 2011; 6683:507-523. Springer, Berlin, Heidelberg. (In Eng.) DOI: https://doi.org/10.1007/978-3-642-25566-3_40
- [3] Thornton C., Hutter F., Hoos H.H., Leyton-Brown K. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13)*. Association for Computing Machinery, New York, NY, USA; 2013. p. 847-855. (In Eng.) DOI: <https://doi.org/10.1145/2487575.2487629>
- [4] Claesen M., De Moor B. Hyperparameter Search in Machine Learning. *arXiv:1502.02127*. 2015. (In Eng.)
- [5] Larsen J., Hansen L.K., Svarer S., Ohlsson M. Design and regularization of neural networks: the optimal use of a validation set. In: *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*, Kyoto, Japan; 1996. p. 62-71. (In Eng.) DOI: <https://doi.org/10.1109/NNSP.1996.548336>
- [6] Bergstra J., Bardenet R., Bengio Y., Kégl B. Algorithms for hyper-parameter optimization. In: *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS'11)*. Curran Associates Inc., Red Hook, NY, USA; 2011. p. 2546-2554. (In Eng.)
- [7] Hazan E., Klivans A., Yuan Y. Hyperparameter Optimization: A Spectral Approach. *arXiv:1706.00764*. 2018. (In Eng.)
- [8] Feurer M., Springenberg J.T., Hutter F. Initializing bayesian hyperparameter optimization via meta-learning. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press; 2015. p. 1128-1135. (In Eng.)
- [9] Mendoza H., Klein A., Feurer M., Springenberg J.T., Urban M., Burkart M. Towards Automatically-Tuned Deep Neural Networks. In: Hutter F., Kotthoff L., Vanschoren J. (ed.) *Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Springer, Cham; 2019. p. 135-149. (In Eng.) DOI: https://doi.org/10.1007/978-3-030-05318-5_7
- [10] Olson R.S., Bartley N., Urbanowicz R.J., Moore J.H. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. Association for Computing Machinery, New York, NY, USA; 2016. p. 485-492. (In Eng.) DOI: <https://doi.org/10.1145/2908812.2908918>
- [11] Feurer M., Klein A., Eggenberger K., Springenberg J.T., Blum M., Hutter F. Efficient and Robust Automated Machine Learning. In: Cortes C., Lawrence N., Lee D., Sugiyama M., Garnett R. *Advances in Neural Information Processing Systems*. 2015; 28:2962-2970. Curran Associates, Inc. Available at: <https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf> (accessed 14.07.2020). (In Eng.)
- [12] Conn A.R., Scheinberg K., Vicente L.N. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics; 2009. (In Eng.) DOI: <https://doi.org/10.1137/1.9780898718768>
- [13] Gutmann H.-M. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*. 2001; 19(3):201-227. (In Eng.) DOI: <https://doi.org/10.1023/A:1011255519438>
- [14] Burkov E., Lempitsky V. Deep neural networks with box convolutions. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA; 2018. p. 6214-6224. (In Eng.)
- [15] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference*



- on *Computer Vision and Pattern Recognition (CVPR)*. Boston, MA; 2015. p. 3431-3440. (In Eng.) DOI: <https://doi.org/10.1109/CVPR.2015.7298965>
- [16] Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv:1511.07122*. 2016. (In Eng.)
- [17] Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018; 40(4):834-848. (In Eng.) DOI: <https://doi.org/10.1109/TPAMI.2017.2699184>
- [18] Jégou S, Drozdal M., Vazquez D., Romero A., Bengio Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, HI; 2017. p. 1175-1183. (In Eng.) DOI: <https://doi.org/10.1109/CVPRW.2017.156>
- [19] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (ed.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. *Lecture Notes in Computer Science*. 2015; 9351:234-241. Springer, Cham. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-24574-4_28
- [20] Badrinarayanan V, Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017; 39(12):2481-2495. (In Eng.) DOI: <https://doi.org/10.1109/TPAMI.2016.2644615>
- [21] Paszke A., Chaurasia A., Kim S., Culurciello E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv:1606.02147*. 2016. (In Eng.)
- [22] Akinina A.V., Nikiforov M.B., Savin A.V. Multiscale image segmentation using normalized cuts in image recognition on satellite images. In: *2018 7th Mediterranean Conference on Embedded Computing (MECO)*. Budva, 2018. p. 1-3. (In Eng.) DOI: <https://doi.org/10.1109/MECO.2018.8406066>
- [23] Akinin M.V., Akinina A.V., Sokolov A.V., Tarasov A.S. Application of EM algorithm in problems of pattern recognition on satellite images. In: *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. Bar, 2017. p. 1-4. (In Eng.) DOI: <https://doi.org/10.1109/MECO.2017.7977190>
- [24] Nurshazlyn Mohd Aszemi, Dominic P.D.D. Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2019; 10(6):269-278. (In Eng.) DOI: <http://dx.doi.org/10.14569/IJACSA.2019.0100638>
- [25] Wistuba M., Schilling N., Schmidt-Thieme L. M. Hyperparameter Optimization Machines. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Montreal, QC; 2016. p. 41-50. (In Eng.) DOI: <https://doi.org/10.1109/DSAA.2016.12>

*Поступила 14.07.2020; принята к публикации 25.08.2020;
опубликована онлайн 30.09.2020.
Submitted 14.07.2020; revised 25.08.2020;
published online 30.09.2020.*

Об авторах:

Акинина Александра Владимировна, аспирант кафедры электронных вычислительных машин, ФГБОУ «Рязанский государственный радиотехнический университет имени В.Ф. Уткина» (390005, Россия, г. Рязань, ул. Гагарина, д. 59/1), ORCID: <http://orcid.org/0000-0002-2745-9415>, alexandra.sokolova@mail.ru

Никифоров Михаил Борисович, директор НОЦ «СпецЭВМ», доцент кафедры электронных вычислительных машин, ФГБОУ «Рязанский государственный радиотехнический университет имени В.Ф. Уткина» (390005, Россия, г. Рязань, ул. Гагарина, д. 59/1), кандидат технических наук, доцент, член-корреспондент Академии информатизации образования, ORCID: <http://orcid.org/0000-0002-4796-0776>, nikiforov.m.b@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Alexandra V. Akinina, Postgraduate Student of the Department of Electronic Computers, Ryazan State Radio Engineering University named after V.F. Utkin (59/1 Gagarin St., Ryazan 390005, Russia), ORCID: <http://orcid.org/0000-0002-2745-9415>, alexandra.sokolova@mail.ru

Mikhail B. Nikiforov, Director of the SEC “SpecEVM”, Associate Professor of the Department of Electronic Computers, Ryazan State Radio Engineering University named after V.F. Utkin (59/1 Gagarin St., Ryazan 390005, Russia), Ph.D. (Engineering), Associate Professor, Corresponding member of the Academy of Education Informatization, ORCID: <http://orcid.org/0000-0002-4796-0776>, nikiforov.m.b@mail.ru

All authors have read and approved the final manuscript.

