

УДК 004.023

DOI: 10.35330/1991-6639-2023-6-116-95-102

EDN: DVFIYH

Научная статья

Обучение искусственной нейронной сети с использованием гибридного алгоритма оптимизации PSOJaya

Е. М. Казакова

Институт прикладной математики и автоматизации –
филиал Кабардино-Балкарского научного центра Российской академии наук
360000, Россия, г. Нальчик, ул. Шортанова, 89 А

Аннотация. Метод оптимизации роем частиц (PSO – Particle swarm optimization) и Jaya – это эвристические алгоритмы оптимизации, которые используются для поиска оптимальных решений в задачах оптимизации. Каждый из этих методов имеет свои сильные и слабые стороны, и выбор между ними зависит от конкретной задачи оптимизации и требований к производительности. В данной работе предлагается гибридный алгоритм PSO и Jaya для повышения эффективности оптимизации. В этой статье PSO, Jaya, PSOJaya используются в качестве методов обучения искусственной нейронной сети (ИНС) для задачи классификации набора данных Balance Scale. Результаты работы гибридного алгоритма сравниваются с результатами алгоритмов Backpropagation (метод обратного распространения ошибки), PSO, Jaya. В тестовых расчетах алгоритмы сравниваются на основе среднего значения, медианы, стандартного отклонения и «лучшего» минимального значения ошибок после 30 симуляций. Результаты эксперимента показывают, что ИНС, обученная с помощью PSOJaya, имеет лучшую точность, чем обученные с помощью Backpropagation, PSO и Jaya.

Ключевые слова: эвристический алгоритм, оптимизация, метод роя частиц (PSO), Jaya, метод обратного распространения ошибки (Backpropagation), гибридный алгоритм, конвейерная гибридизация, ИНС, классификация

Поступила 01.11.2023, одобрена после рецензирования 04.11.2023, принята к публикации 14.11.2023

Для цитирования. Казакова Е. М. Обучение искусственной нейронной сети с использованием гибридного алгоритма оптимизации PSOJaya // Известия Кабардино-Балкарского научного центра РАН. 2023. № 6(116). С. 95–102. DOI: 10.35330/1991-6639-2023-6-116-95-102

MSC: 68T07

Original article

Training an artificial neural network using the PSOJaya hybrid optimization algorithm

E.M. Kazakova

Institute of Applied Mathematics and Automation –
branch of Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences
360000, Russia, Nalchik, 89 A Shortanov street

Abstract. Particle swarm optimization (PSO) and Jaya are heuristic optimization algorithms that are used to find optimal solutions in optimization problems. Each of these methods has its own advantages and disadvantages, and the choice between them depends on the specific optimization problem and performance requirements. This paper proposes a hybrid of PSO and Jaya algorithms to improve optimization efficiency. In this paper PSO, Jaya, PSOJaya are used as artificial neural network (ANN) training methods for the classification task on the Balance Scale dataset. The results of the hybrid algorithm are compared with the

results of the Backpropagation, PSO and Jaya algorithms. The test calculations compare the algorithms based on the mean, median, standard deviation, and "best" minimum error value after 30 simulations. The experiment results show that the ANN trained with PSOJaya has higher accuracy than the one trained with Backpropagation, PSO and Jaya.

Keywords: heuristic algorithm, optimization, particle swarm method (PSO), Jaya, Backpropagation, hybrid algorithm, pipeline hybridization, ANN, classification

Submitted 01.11.2023,

approved after reviewing 04.11.2023,

accepted for publication 14.11.2023

For citation. Kazakova E.M. Training an artificial neural network using the PSOJaya hybrid optimization algorithm. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2023. No. 6(116). Pp. 95–102. DOI: 10.35330/1991-6639-2023-6-116-95-102

ВВЕДЕНИЕ

PSO – это метод оптимизации, используемый в вычислительной математике и искусственном интеллекте для поиска оптимальных значений в задачах оптимизации. PSO моделирует процесс оптимизации как движение «частиц» в пространстве поиска, где каждая частица представляет собой потенциальное решение задачи, и они сотрудничают, чтобы найти оптимальное решение, подражая поведению стаи птиц или группы животных [1].

Алгоритм Jaya (Jaya в переводе с санскрита означает «victory» (победа), также называемый «джая») – это оптимизационный алгоритм, который был разработан для решения задач оптимизации [2]. Jaya является относительно простым и эффективным алгоритмом, используемым для нахождения оптимальных значений в задачах оптимизации без градиентов и ограничений. Основная идея данного алгоритма состоит в том, что он использует популяцию решений и итеративно улучшает их, двигаясь к лучшим решениям в пространстве решений. Однако Jaya не всегда стабильно сходится к оптимальным решениям и может иметь переменное качество оптимизации в зависимости от задачи и параметров.

Выбор алгоритма оптимизации зависит от характеристик конкретной задачи, однако имеющийся эвристический алгоритм может быть гибридизирован с другими эвристическими алгоритмами с целью улучшения их производительности и способности решать сложные оптимизационные задачи. Путем объединения разных методов гибридные алгоритмы могут комбинировать преимущества каждого алгоритма и справляться с ограничениями и сложностями, с которыми они сталкиваются. Гибридизация методов оптимизации, таких как PSO и Jaya, может быть полезной для достижения преимуществ обоих методов и преодоления их ограничений.

Существует несколько подходов к созданию гибридных эвристических алгоритмов [3]: вложение (embedded) [4, 5], препроцессор/постпроцессор, включающий последовательную и конвейерную гибридизацию (preprocessor/postprocessor) [6], коалгоритмы (coalgorithms) [7].

В данной работе разработан гибридный алгоритм PSOJaya по принципу конвейерной гибридизации без изменения общей работы алгоритмов PSO и Jaya.

Конвейерная гибридизация предполагает последовательное выполнение поисковых алгоритмов оптимизации, пока не будет достигнут критерий останова.

АЛГОРИТМ PSO

Метод PSO направлен на итеративную оптимизацию задачи, начиная с набора или популяции возможных решений, называемых роем частиц, в котором каждая частица знает глобальное наилучшее положение в рое, а также его индивидуальное наилучшее положение, найденное до сих пор в процессе поиска в пространстве решения задачи.

Вначале инициализируется рой частиц x_i из N D -мерных векторов с действительными значениями и аналогично генерируется скорость частиц v_i каждой частицы. После инициализации роя и скорости для каждой частицы вычисляются значение целевой функции и

начальное индивидуальное лучшее положение $f(p_{best_i}^0)$ вместе с начальным глобальным (или окрестным) наилучшим положением $f(g_{best}^0)$. На следующем этапе каждая частица обновляет свою скорость и положение, исходя из ее текущего положения, скорости и опыта, полученного от лучшего решения в популяции по формулам

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (1)$$

$$v_{i,j}^{t+1} = \omega v_{i,j}^t + c_1 r_{1,i,j}^{t+1} (p_{best,i,j}^t - x_{i,j}^t) + c_2 r_{2,i,j}^{t+1} (g_{best,j}^t - x_{i,j}^t), \quad (2)$$

где $v_{i,j}^0 = 0$ и $v_{i,j}^{t+1}$ – вектор скорости i -й частицы на итерации $t+1$ в j измерении, $p_{best,i,j}^t$ – наилучшее индивидуальное положение частицы, $g_{best,j}^t$ – наилучшее глобальное положение среди всех частиц в рое, $r_{1,i,j}^{t+1}$, $r_{2,i,j}^{t+1}$ – случайные величины, принимающие значения на текущей итерации $t+1$ из диапазона $[0, 1]$, ω (инерциальный вес), c_1 , c_2 – параметры алгоритма. Первое слагаемое в уравнении (2) называется инерционной компонентой, которая обеспечивает частице соответствующий импульс для перемещения по пространству поиска без резкого изменения направления на основе истории предыдущих направлений полета, второе слагаемое называется когнитивной компонентой и обозначает память о предыдущем индивидуальном лучшем положении, в котором получала свою лучшую результативность в прошлом; третье слагаемое – социальная компонента, определяющая текущую производительность, связанную с лучшими глобальными решениями, найденными до сих пор.

АЛГОРИТМ JAYA

Основная идея алгоритма Jaya заключается в том, что каждое решение в популяции должно стремиться быть лучше, и для этого оно может быть обновлено на основе лучших решений, найденных в популяции. Это позволяет алгоритму исследовать пространство решений и находить лучшие решения по мере продвижения по итерациям. Вначале генерируется популяция решений x_i , на следующем этапе вычисляется значение целевой функции $f(x_i)$. Среди всех полученных решений-кандидатов отбираются частица x_{best} с наилучшим положением и соответствующее наилучшее значение f_{best} , а наихудший кандидат x_{worst} получает наихудшее значение f_{worst} . Каждое решение $f(x_{k,j}^i)$ ($x_{k,j}^i$ – k -я частица с j -й переменной на i -й итерации) в популяции сравнивается с текущим наилучшим решением $f(x_{best,j}^i)$ ($x_{best,j}^i$ – наилучшая частица с j -й переменной на i -й итерации). Если $f(x_{k,j}^i) < f(x_{best,j}^i)$ ($f(x_{k,j}^i) > f(x_{best,j}^i)$), то текущее положение частицы обновляется по формуле

$$x_{k,j}^{i'} = x_{k,j}^i + r_{1,j}^i (x_{best,j}^i - |x_{k,j}^i|) - r_{2,j}^i (x_{worst,j}^i - |x_{k,j}^i|), \quad (3)$$

где $r_{1,j}^i$, $r_{2,j}^i$ – случайные величины, принимающие значения на текущей итерации i из диапазона $[0, 1]$, $x_{best,j}^i$ – значение переменной j для наилучшего кандидата, а $x_{worst,j}^i$ – значение переменной j для наихудшего кандидата, $x_{k,j}^{i'}$ – обновленное значение $x_{k,j}^i$. Член $r_{1,j}^i (x_{best,j}^i - |x_{k,j}^i|)$ указывает на стремление решения приблизиться к лучшему решению, а член $r_{2,j}^i (x_{worst,j}^i - |x_{k,j}^i|)$ указывает на тенденцию решения избегать наихудшего решения. Все принятые значения функции в конце итерации сохраняются, и эти значения становятся входными данными для следующей итерации.

ГИБРИДНЫЙ АЛГОРИТМ PSO-JAYA

Гибридный алгоритм PSOJaya является комбинацией двух оптимизационных алгоритмов – PSO и Jaya. Преимущество гибридного метода заключается в увеличении скорости сходимости, так как оба метода могут взаимодополнять друг друга и помогать ускорить процесс оптимизации, также он может способствовать большей разнообразности поиска за счет различных подходов в стратегии обновления решений, что может быть полезно в задачах, где разнообразие решений важно. В случае, если один из методов сталкивается с трудностью в оптимизации, другой метод может компенсировать его недостатки.

Общий принцип работы гибридного алгоритма PSOJaya:

1. Инициализация: создается начальная популяция частиц размером M для PSO с пунктом 1 раздела PSO.
2. PSO-фаза: частицы PSO итеративно обновляют свое положение и скорость, исследуя пространство решений. Целевая функция вычисляется для каждой частицы, обновляются лучшие решения, найденные каждой частицей и во всей популяции (раздел PSO).
3. Jaya-фаза: после завершения PSO-фазы в алгоритм Jaya передается N ($N \leq M$) лучших найденных решений алгоритмом PSO из M решений. Используя алгоритм Jaya, выполняется итеративное обновление решений по формуле (3) с целью улучшения их качества (раздел Jaya). После завершения работы Jaya обновляются положения N частиц в PSO из Jaya.
4. Проверка критерия останова: алгоритм продолжает выполнять PSO-фазы и Jaya-фазы, пока не будет достигнут заданный критерий останова, такой как максимальное количество итераций или требуемая точность решения.

ОБУЧЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ

Существует ряд методов использования эвристических алгоритмов для обучения нейросети:

1. Эвристические алгоритмы используются для нахождения комбинации весов и смещений, обеспечивающей минимальную ошибку для ИНС [8].
2. Эвристические алгоритмы используются как способ найти правильную структуру для ИНС в конкретной задаче [9].
3. Эвристические алгоритмы используются для настройки параметров алгоритма обучения на основе градиента [10].

В данной статье PSO, Jaya и PSOJaya применяются к ИНС с использованием первого метода. Это означает, что структура ИНС фиксирована, алгоритмы находят комбинацию весов и смещений, которая дает минимальную ошибку для ИНС. Для проектирования ИНС необходимо определить основные элементы: во-первых, следует определить функцию потерь, использующую ошибку ИНС, для оценки пригодности частиц в ИНС с алгоритмами обучения PSO, Jaya и PSOJaya; во-вторых, необходимо определить стратегию кодирования весов и смещений для каждого ИНС с соответствующими алгоритмами.

ФУНКЦИЯ ПОТЕРЬ

Архитектура ИНС в данной статье имеет вид: один входной, один скрытый и один выходной слои, где количество нейронов на входном слое равно n , на скрытом слое – h , а на выходном слое – m ($n - h - m$). Функцией активации служит сигмоида. После вычисления нейронов на выходном слое ошибка сети (функция стоимости) будет иметь вид

$$E = \frac{1}{q} \sum_{i=1}^m (\widehat{y}_i^k - y_i^k)^2,$$

где q – количество экземпляров в обучающем наборе, \widehat{y}_i^k – выход сети для k -го экземпляра, y_i^k – метка для k -го экземпляра в обучающем наборе.

КОДИРОВАНИЕ ВЕСОВ

После определения функции пригодности следующим этапом является выбор стратегии кодирования для представления весов и смещений ИНС для каждой частицы в PSO, Jaya и PSOJaya. Согласно [11], существует три метода кодирования и представления весов и смещений ИНС для каждой частицы в эвристических алгоритмах. В данной статье использовалась стратегия векторного кодирования. Для задействованной ИНС каждая частица представляет все веса и смещения структуры нейросети. Например, для ИНС со структурой 2 входа, 4 нейрона на скрытом слое, 1 выход (2–4–1) можно представить соответствующее кодирование для каждой частицы как:

$$particle(i) = [w_{11}, w_{12}, w_{13}, w_{14}, w_{21}, w_{22}, w_{23}, w_{24}, \overline{w_{11}}, \overline{w_{21}}, \overline{w_{31}}, \overline{w_{41}}],$$

$$particle_matrix = [particle(1), particle(2), \dots, particle(M)],$$

где $w_{11}, w_{12}, w_{13}, w_{14}, w_{21}, w_{22}, w_{23}, w_{24}$ – веса между входом и скрытым слоем, $\overline{w_{11}}, \overline{w_{21}}, \overline{w_{31}}, \overline{w_{41}}$ – веса между скрытым слоем и выходом, M – общее количество частиц (рой), $i=1, \dots, M$. При вычислении выходных данных ИНС частицы снова декодируются в матрицу весов, поэтому процесс декодирования становится немного сложным. Тем не менее такая стратегия кодирования часто используется в области оптимизации функций.

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Для сравнения эффективности работы алгоритмов Backpropagation (BP), PSO, Jaya и PSOJaya в обучении ИНС проведен вычислительный эксперимент с ИНС по классификации состояний объекта на наборе данных Balance Scale (шкалы баланса). Набор данных получен из UC Irvine Machine Learning Repository [12]. Экземпляры в наборе данных соответствуют одному из 3 классов: шкала справа, шкала слева или шкала сбалансирована. Набор данных состоит из 625 экземпляров и 4 признаков: левый вес, левое расстояние, правый вес и правое расстояние. Например, элемент набора данных «2, 5, 2, 1» (левый вес = 2, левое расстояние = 5, правый вес = 2, правое расстояние = 1) следует классифицировать по классу «слева». Для обучения используется 468 образцов, а для тестирования – 157. Архитектура нейросети имеет вид 4–18–3.

В этой задаче предполагается, что каждая частица инициализируется случайным образом в диапазоне $[-1, 1]$. Алгоритмы имеют следующие параметры: для PSO, PSO-Jaya $\omega = 2$, $c_1 = 1$, $c_2 = 0,5$, $v_{max} = 0,1$. Размер роя для трех алгоритмов $M = 40$, в гибридном алгоритме из M частиц роя в Jaya передается $N = 10$ наилучших найденных решений PSO.

Для оценки эффективности работы классификатора используются метрики *accuracy* (верность) и F_1 .

Алгоритмы BP, PSO, Jaya и PSOJaya сравниваются на основе среднего значения, медианы, стандартного отклонения и наилучшей из среднеквадратичной ошибки (MSE – Mean Square Error) для обучающего набора Balance scale после 30 симуляций. Критерием завершения процесса обучения является выполнение максимального количества итераций (в данном случае 500). Экспериментальные результаты для задачи приведены в таблицах 1, 2. Наилучшие результаты выделены жирным шрифтом.

В таблице 1 указаны значения F_1 -метрик и *accuracy* для классификатора с алгоритмами BP, PSO, Jaya и PSOJaya после 30 симуляций. Как видно из таблицы, гибридный алгоритм PSOJaya превзошел алгоритмы PSO, Jaya, достигнув лучшей точности классификации на обучающем и тестовом наборах.

Таблица 1. Среднее значение метрик *accuracy* и F_1 после 30 симуляций**Table 1.** Average accuracy and F_1 metric after 30 simulations

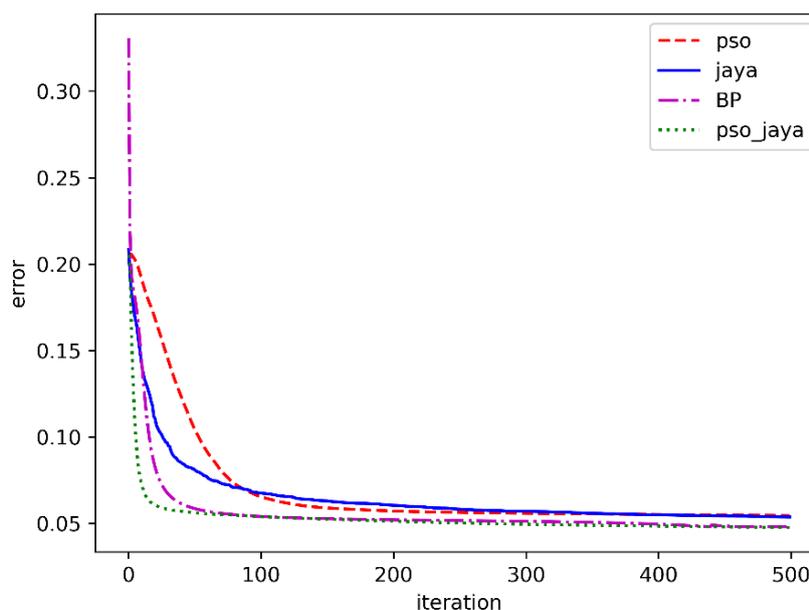
№	Алгоритм	На обучающем наборе (%)	На тестовом наборе (%)	F_1 метрика
1	BP	90,58	91,67	0,91
	PSO	90,28	91,3	0,91
	Јауа	88,78	89,36	0,89
	PSO-Јауа	90,99	92,52	0,92

Из таблицы 2 видно, что классификатор с гибридным алгоритмом работает лучше, чем другие классификаторы с алгоритмами PSO, Јауа, исходя из значений среднего, медианы, стандартного отклонения и минимального MSE. Эти статистические данные показывают, что классификатор с PSOЈауа имеет лучшую способность избегать локальных минимумов. Также для минимального MSE гибридный алгоритм имеет лучшие результаты, что указывает на большую точность классификатора, чем классификаторов с алгоритмами PSO и Јауа.

Таблица 2. Среднее, медиана, стандартное отклонение и best MSE**Table 2.** Mean, median, standard deviation and best MSE

№	Алгоритм	Среднее	Медиана	Станд. отклонение	best MSE
1	BP	0,0449	0,0448	1,3877e-17	0,0448
	PSO	0,0581	0,0575	0,0016	0,0549
	Јауа	0,0615	0,0606	0,0056	0,0517
	PSO-Јауа	0,0486	0,0485	0,0016	0,046

На рисунке 1 показана кривая сходимости классификатора данных Balance Scale с BP, PSO, Јауа и PSOЈауа, основанная на средних значениях MSE для обучающего набора данных после 30 симуляций. Кривая показывает, что гибридный алгоритм PSOЈауа дает лучшую скорость сходимости для классификатора данных Balance scale.

**Рис. 1.** Среднее значение MSE**Fig. 1.** Average MSE value

ЗАКЛЮЧЕНИЕ

В данной статье представлен гибридный алгоритм для задач оптимизации на основе двух эвристических алгоритмов PSO и Jaya. Идея гибридного алгоритма заключается в конвейерной последовательной работе двух алгоритмов PSO и Jaya. Чтобы оценить эффективность работы предложенного гибридного алгоритма, было произведено обучение нейросети для задачи классификации Balance scale.

На основе полученных данных в результате экспериментов можно заключить, что гибридный алгоритм PSOJaya дает лучшие результаты, чем алгоритмы PSO, Jaya по отдельности и BP для рассматриваемой задачи классификации.

REFERENCES

1. Kennedy J., Eberhart R. Particle Swarm Optimization. *IEE International Conference on Neural Networks*. 1995. Pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968
2. Rao R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Indus Eng Comput*. 2016. Vol. 1. No. 7. Pp. 19–34. DOI: 10.5267/j.ijiec.2015.8.004
3. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. 2 издание. М.: МГТУ им. Н. Э. Баумана, 2017. 446 с.
Karpenko A.P. *Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy* [Modern search engine optimization algorithms. Algorithms inspired by nature]. Moscow: MGTU im. N. E. Bauman, 2017. 446 p. (In Russian)
4. Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation*. 2016. Vol. 274. Pp. 292–305. DOI: 10.1016/j.amc.2015.11.001
5. Mirjalili S., Hashim S.-Z.M. A new hybrid PSOGSA algorithm for function optimization. *Proceedings of ICCIA 2010-2010 International Conference on Computer and Information Application*. 2010. Pp. 374–377. DOI: 10.1109/ICCIA.2010.6141614
6. Şenel F.A., Gökçe F. Yüksel A.S. et al. A novel hybrid PSO–GWO algorithm for optimization problems. *Engineering with Computers*. 2019. Vol. 35. Pp. 1359–1373. DOI: 10.1007/s00366-018-0668-5
7. Zhou Y., Shengyu P.A. Hybrid Co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. *J. Comput*. 2010. Vol. 6. No. 5. Pp. 965–972. DOI: 10.4304/jcp.5.6.965-972
8. Mirjalili S.A., Hashim S.-Z.M., Sardroudi H.M. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*. 2012. Vol. 22. No. 218. Pp. 11125–11137. DOI: 10.1016/j.amc.2012.04.069
9. Junior F.-E.F., Yen G.G. Particle swarm optimization of deep neural networks architectures for image classification. *Swarm and Evolutionary Computation*. Vol. 49. 2019. Pp. 62–74. DOI: 10.1016/j.swevo.2019.05.010
10. Garro B.A., Vázquez R.A. Designing artificial neural networks using particle swarm optimization algorithms. *Computational intelligence and neuroscience*. 2015. P. 61. DOI: 10.1155/2015/369298
11. Zhang J.R., Zhang J., Lok T.M., Lyu M.R. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied mathematics and computation*. 2007. Vol. 2. No. 185. Pp. 1026–1037.

12. Siegler R. Balance Scale. UCI Machine Learning Repository. 1994. <https://doi.org/10.24432/C5488X>.

Информация об авторе

Казакова Елена Мусовна, мл. науч. сотр. отдела нейроинформатики и машинного обучения, Институт прикладной математики и автоматизации – филиал Кабардино-Балкарского научного центра Российской академии наук;

360000, Россия, г. Нальчик, ул. Шортанова, 89 А;

shogenovae@inbox.ru, ORCID: <https://orcid.org/0000-0002-5819-9396>

Information about the author

Kazakova Elena Musovna, Junior Researcher of the Department of Neuroinformatics and Machine Learning, Institute of Applied Mathematics and Automation – branch of Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences;

360000, Russia, Nalchik, 89 A Shortanov street;

shogenovae@inbox.ru, ORCID: <https://orcid.org/0000-0002-5819-9396>