

УДК: 004.032.26

DOI: <https://doi.org/10.47813/2782-2818-2024-4-2-0291-0297>

EDN: STJEYM



## Использование ройного интеллекта для оптимизации гиперпараметров нейронных сетей: сравнительный анализ на MNIST и CIFAR-10

А. А. Инкижеков, А. С. Дулесов

*Хакасский государственный университет им. Н.Ф. Катанова, Абакан, Россия*

**Аннотация.** Ройный интеллект предлагает мощные методы для решения задач оптимизации, применяемые в настройке гиперпараметров нейронных сетей. В данной статье исследуется производительность алгоритма ройной оптимизации частиц по сравнению с перебором по сетке на двух различных датасетах: MNIST и CIFAR-10. Экспериментальные результаты показывают, что эффективность методов оптимизации варьируется в зависимости от сложности задачи и данных.

**Ключевые слова:** ройный интеллект, оптимизация гиперпараметров, нейронные сети, алгоритм ройной оптимизации частиц, перебор по сетке, MNIST, CIFAR-10, эффективность алгоритмов, методы машинного обучения.

**Для цитирования:** Инкижеков А. А., & Дулесов, А. С. (2024). Использование ройного интеллекта для оптимизации гиперпараметров нейронных сетей: сравнительный анализ на MNIST и CIFAR-10. *Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies*, 4(2), 0291–0297. <https://doi.org/10.47813/2782-2818-2024-4-2-0291-0297>

## Using swarm intelligence to optimize neural network hyperparameters: comparative analysis on MNIST and CIFAR-10

A. A. Inkizhekov, A. S. Dulesov

*Katanov Khakass State University, Abakan, Russia, Country*

**Abstract.** Swarm Intelligence offers powerful methods for solving optimization problems used in configuring hyperparameters of neural networks. This article examines the performance of the particle swarm optimization algorithm compared to Grid Search on two different datasets: MNIST and CIFAR-10. Experimental results show that the effectiveness of optimization methods varies depending on the complexity of the task and the data.

**Keywords:** swarm intelligence, hyperparameter optimization, neural networks, particle swarm optimization, grid search, MNIST, CIFAR-10, algorithm efficiency, machine learning methods.

**For citation:** Inkizhekov A. A., & Dulesov, A. S. (2024). Using swarm intelligence to optimize neural network hyperparameters: comparative analysis on MNIST and CIFAR-10. *Modern Innovations, Systems and Technologies*, 4(2), 0291–0297. <https://doi.org/10.47813/2782-2818-2024-4-2-0291-0297>

## ВВЕДЕНИЕ

Оптимизация гиперпараметров нейронных сетей является критическим аспектом в машинном обучении, существенно влияющим на производительность моделей. Grid Search (поиск по сетке гиперпараметров), традиционный метод, известен своей простотой, но требует значительных вычислительных ресурсов [1]. Алгоритмы ройного интеллекта, такие как PSO (particle swarm optimization, ройная оптимизация частиц), предлагают альтернативные подходы, которые могут ускорить поиск оптимальных решений, особенно при увеличении числа параметров [2].

Цель данной работы – провести сравнительный анализ PSO и Grid Search для оптимизации гиперпараметров нейронных сетей на датасетах разной сложности: MNIST и CIFAR-10. Мы исследуем, как эти методы адаптируются к различным условиям и какие преимущества предоставляет каждый из них.

## МАТЕМАТИЧЕСКИЕ ОСНОВЫ РОЙНОГО ИНТЕЛЛЕКТА

Методы ройного интеллекта, такие как PSO, вдохновлены коллективным поведением живых существ, таких как стаи птиц и косяки рыб. В PSO каждая "частица" представляет собой потенциальное решение, которое перемещается в пространстве поиска в соответствии со своим собственным опытом и опытом соседей. Алгоритм ройной оптимизации частиц (PSO) моделирует коллективное поведение роя частиц, которые перемещаются в пространстве поиска с учетом как личного опыта, так и глобальных знаний роя [2]. Каждая частица обновляет свою позицию и скорость в поиске оптимального решения, минимизируя заданную функцию потерь.

### Математическая формулировка PSO

$$v_i(t + 1) = \omega v_i(t) + \phi_p r_p (p_i - x_i(t)) + \phi_g r_g (g - x_i(t))$$

где:

- $v_i(t)$  - скорость частицы  $i$  в момент времени  $t$ ,

- $x_i(t)$  - текущее положение частицы  $i$ ,
- $p_i$  - лучшее индивидуальное положение частицы  $i$ ,
- $g$  - лучшее глобальное положение, найденное роем,
- $\omega$  - коэффициент инерции,
- $\phi_p$  и  $\phi_g$  - коэффициенты ускорения для личного и глобального опыта,
- $r_p$  и  $r_g$  - случайные числа, равномерно распределенные между 0 и 1.

### Обучение с подкреплением с использованием ройных частиц (RL-PSO)

Обучение с подкреплением (RL) в сочетании с PSO может значительно повысить эффективность оптимизации за счет адаптивного выбора стратегии обновления параметров. В RL-PSO каждая частица является агентом, который обучается с использованием правил подкрепления для улучшения своих движений в пространстве поиска. Это достигается путем награждения частиц за нахождение лучшего решения и штрафования за худшие решения.

Математически это можно описать следующим образом:

1. Функция награды  $R(x_i)$ , которая оценивает качество текущего решения  $x_i$

$$R(x_i) = -f(x_i).$$

2. Функция обновления политики  $\pi(x_i)$ , которая направляет частицы на улучшение их решений:

$$\pi(x_i) = x_i + \alpha \nabla R(x_i),$$

где  $\alpha$  – коэффициент обучения, а  $\nabla R(x_i)$  – градиент награды.

### Применение RL-PSO в оптимизации нейронных сетей

RL-PSO применяется для настройки гиперпараметров и архитектуры нейронных сетей. В данном методе каждая частица представляет собой набор гиперпараметров и архитектурных решений, таких как количество слоев, количество нейронов в каждом слое, коэффициенты регуляризации и скорость обучения.

Приведем следующие примеры использования RL-PSO.

- Инициализация роя частиц, где каждая частица кодирует возможные гиперпараметры и архитектуру сети.

- Оценка эффективности текущей сети (частицы) с помощью функции потерь.
- Применение RL-обучения для обновления гиперпараметров, основываясь на градиенте функции потерь.
- Обновление скорости и положения частиц на основе функции обновления политики.
- Повторение шагов 2-4 до достижения критерия остановки.

Применительно к задаче классификации на CIFAR-10, такая настройка может включать:

- количество сверточных и полносвязных слоев,
- количество фильтров и размер ядра,
- коэффициенты L2-регуляризации и dropout,
- скорость обучения.

### Экспериментальная установка

Мы сравниваем PSO и Grid Search на двух датасетах:

- MNIST: Набор данных из рукописных цифр, представляющий относительно простую задачу классификации [5].
- CIFAR-10: Набор данных из цветных изображений, представляющий более сложную задачу классификации [6].

Для каждой задачи мы оптимизируем следующие гиперпараметры:

- количество нейронов (*num\_neurons*): 64, 128, 256;
- скорость обучения (*learning\_ratelearning*): 0.001, 0.01, 0.1;
- коэффициент исключения (*dropout\_rate*): 0.2, 0.4, 0.6;
- коэффициент L2-регуляризации (*l2\_ratel2*): 0.001, 0.005, 0.01;
- количество слоев (*num\_layersnum*): 1, 2, 3.

В качестве метрики оценки рассматривается:

- точность модели на проверочных данных;
- время выполнения для оптимизации гиперпараметров.

### РЕЗУЛЬТАТЫ

В таблицах 1 и 2 представлены результаты для MNIST и CIFAR-10.

Таблица 1. Результаты для MNIST.

Table 1. Results for MNIST.

Метод	Время (сек)	Лучшая точность
Grid Search	50.123	0.9764
PSO	120.678	0.9642

На MNIST Grid Search показал лучшие результаты как по точности, так и по времени выполнения. Это объясняется относительно небольшой размерностью пространства параметров и простотой задачи классификации, что делает перебор по сетке более эффективным [5].

Таблица 2. Результаты для CIFAR-10.

Table 2. Results for CIFAR-10.

Метод	Время (сек)	Лучшая точность
Grid Search	534.239	0.2935
PSO	132.270	0.3095

На CIFAR-10 PSO показал лучшее время выполнения и немного более высокую точность по сравнению с Grid Search. Это можно объяснить более сложной природой задачи классификации изображений CIFAR-10 и большей размерностью пространства параметров, где ройный интеллект может эффективнее исследовать сложные области поиска [6].

### Анализ результатов

1. Простота задачи и размерность пространства: Grid Search более эффективен на простых задачах с небольшой размерностью пространства параметров, таких как MNIST [5].
2. Сложность задачи и многообразие параметров: PSO показывает лучшие результаты на сложных задачах с большим числом параметров, таких как CIFAR-10, поскольку он способен лучше справляться с высокоразмерными и сложными пространствами поиска [6].

Ссылка на примеры экспериментальных установок: сравнение MNIST и CIFAR-10 - [https://github.com/Inkizhekov/MNIST\\_-\\_CIFAR-10\\_compare\\_swarm](https://github.com/Inkizhekov/MNIST_-_CIFAR-10_compare_swarm) .

Следует учесть, что время оптимизации параметров может существенно отличаться в зависимости от конфигурации тестового компьютера, на котором будут исполняться

примеры. Также следует учесть, что посев частиц в PSO происходит случайным образом. И время поиска оптимальных гиперпараметров может отличаться от приведенных в статье. Но общая тенденция при росте размерности оптимизируемого пространства гиперпараметров сохраняется при многократном повторении оптимизации.

### **Перспективы и дальнейшие исследования**

Преимущества RL-PSO заключаются в его способности адаптироваться к различным пространствам параметров и архитектурам сети. В будущем возможны улучшения за счет интеграции с другими методами оптимизации и более глубокого использования RL для динамического выбора стратегий поиска. Кроме того, комбинация PSO с генетическими алгоритмами или алгоритмами дифференциальной эволюции может привести к улучшенным результатам в более сложных сценариях [1-4].

Исследования также могут сосредоточиться на применении RL-PSO в реальных задачах машинного обучения, таких как обработка изображений, обработка естественного языка и задачи большого масштаба, где традиционные методы могут быть менее эффективны.

### **ЗАКЛЮЧЕНИЕ**

В статье рассмотрены теоретические и практические аспекты использования методов ройного интеллекта, таких как RL-PSO, для оптимизации гиперпараметров и архитектуры нейронных сетей. Проведено сравнение с перебором по сетке на примере датасетов MNIST и CIFAR-10. Показано, что RL-PSO эффективен в задачах с большим количеством гиперпараметров и сложными архитектурами сетей, в то время как Grid Search лучше подходит для задач с небольшой размерностью параметров. В будущем комбинированные подходы могут предоставить более универсальные решения для различных условий [1-6].

Алгоритмы ройного интеллекта, такие как PSO, и традиционные методы, такие как Grid Search, имеют свои преимущества и ограничения в зависимости от конкретных условий задачи. Наши эксперименты показали, что Grid Search более эффективен для простых задач с небольшой размерностью пространства параметров, тогда как PSO превосходит его в сложных задачах с большим числом гиперпараметров, требующих более эффективного поиска. В будущем исследование должно сосредоточиться на

комбинированных методах, которые могут адаптироваться к различным условиям и эффективно использовать преимущества обоих подходов [1-4].

### СПИСОК ЛИТЕРАТУРЫ

- [1] Kennedy J., Eberhart R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. 1995; 4: 1942-1948. DOI: 10.1109/ICNN.1995.488968
- [2] Poli R., Kennedy J., Blackwell T. Particle swarm optimization: An overview. Swarm Intelligence. 2007; 1(1): 33-57. DOI: 10.1007/s11721-007-0002-0
- [3] Shi Y., Eberhart R.C. A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence. 1998; Cat.No.98TH8360: 69-73. DOI: 10.1109/ICEC.1998.699146
- [4] Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press; 2016.
- [5] LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11): 2278-2324. DOI: 10.1109/5.726791
- [6] Krizhevsky A., Hinton G. Learning multiple layers of features from tiny images. Technical Report. University of Toronto; 2009. Link. <http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>

### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Инкижеков Анатолий Анатольевич**, аспирант, кафедры цифровых технологий и дизайна, Хакасский государственный университет имени Н.Ф. Катанова, Абакан, Россия

**Anatolii Inkizhekov**, PhD student at the Faculty of Digital Technologies and Design of N.F. Katanov Khakass State University, Abakan, Russia

**Дулесов Александр Сергеевич**, доктор технических наук, доцент, профессор кафедры цифровых технологий и дизайна, Хакасский государственный университет имени Н.Ф. Катанова, Абакан, Россия

**Alexander Dulesov**, Doctor of Technical Sciences, Associate Professor, Professor of the Department of Digital Technologies and Design, Katanov Khakass State University, Abakan, Russia

*Статья поступила в редакцию 14.06.2024; одобрена после рецензирования 26.06.2024; принята к публикации 27.06.2024.*

*The article was submitted 14.06.2024; approved after reviewing 26.06.2024; accepted for publication 27.06.2024.*