

УДК 004.932.75'1+004.89

 10.25209/2079-3316-2022-13-3-225-239



Кластеризация рукописных цифр нейронной сетью Кохонена

Дина Сергеевна Латыпова^{1✉}, Дмитрий Николаевич Тумаков²

^{1,2} Казанский (Приволжский) федеральный университет, Казань, Россия

[✉] dina.latyova23@gmail.com

(подробнее об авторах на с. 239)

Аннотация. Проведена кластеризация рукописных цифр по шестидесяти тысячам изображений, содержащихся в обучающей выборке базы данных MNIST. Для кластеризации использована нейронная сеть Кохонена. Для каждой рукописной цифры определено оптимальное количество кластеров (не более 50). При определении расстояния между объектами (изображениями рукописных цифр) использована евклидова норма. Проверка правильности построения кластеров проведена по данным из тестовой выборки базы MNIST. Тестовая выборка содержит десять тысяч изображений. Сделан вывод о том, что изображения из тестовой выборки принадлежат кластеру «правильной цифры» с вероятностью более 90%. Для каждой цифры вычислена F-мера для оценки кластеров. Наилучшие значения F-меры получены для цифр 0 и 1 (F-среднее значение равно 0,974). Наихудшие значения получены для цифры 9 (F-среднее значение равно 0,903). Также проведён кластерный анализ, который позволил сделать выводы о возможных ошибках в распознавании нейронной сетью Кохонена. Построены пересечения кластеров для изображений рукописных цифр. Приведены примеры пересечений кластеров, а также показаны примеры изображений, которые распознаны нейронной сетью неверно.

Ключевые слова и фразы: нейронная сеть Кохонена, кластеризация, MNIST

Благодарности:

^{1,2} Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»)

Для цитирования: Латыпова Д.С., Тумаков Д.Н. *Кластеризация рукописных цифр нейронной сетью Кохонена* // Программные системы: теория и приложения. 2022. Т. 13. № 3(54). С. 225–239. http://psta.psiras.ru/read/psta2022_3_225-239.pdf



Введение

Распознавание рукописного текста представляет собой сложную задачу, которая в данный момент не решена в полном объеме. Использование искусственного интеллекта при распознавании рукописного текста значительно ускоряет процесс его обработки. Одна из проблем в распознавании образов нейронными сетями связана с категоризацией данных. Данные в неподходящей категории означают неверную информацию, что влечёт за собой ошибки. Кластеризация образов является одним из эффективных подходов к решению данной проблемы [1].

Распознавание образов выступает в качестве основного шага для обеспечения кластеризации, поскольку этот процесс анализирует структуру и векторное значение каждого символа в наборе данных. Например, в [2] предложен алгоритм консенсусной кластеризации для набора рукописных цифр. В [3] описана надежная непрерывная кластеризация. В [4] предложен алгоритм для улучшения эффективности классификации при распознавании рукописных цифр. Для кластеризации рукописных цифр нередко используются подходы *s-means* [5] и *k-means* [6, 7], а также различные нейронные сети. Например, в [8] рукописные буквы сгруппированы с использованием нейронной сети Кохонена. В [9] описаны разные подходы для кластеризации: кластеризация на основе разделов, иерархическая кластеризация, кластеризация на основе плотности и кластеризация на основе сетки. Один из самых популярных алгоритмов кластеризации, *k-means*, описан в [10]. При использовании этого метода достигается точность примерно в 90%. В случае больших измерений также может быть использована кластеризация подпространства [11, 12].

В настоящей работе с использованием нейронной сети Кохонена выполнена кластеризация рукописных цифр из базы данных MNIST. Основой для нашего исследования стали результаты статьи [13], в которой проведена кластеризация рукописных цифр, для каждой цифры выделено по три кластера и показано, что из-за больших размеров кластеров цифры из тестовой выборки имеют невысокое попадание в «свои» кластеры. Ниже количество кластеров для каждой цифры выбрано разным, но не превышающим 50. Ограничение количества кластеров числом 50 обусловлено дальнейшим использованием результатов кластеризации для нейронной сети Хопфилда, которая имеет ограничение на количество входных объектов (кластеров)

[14]. Для оценки расстояния между изображениями цифр возможно использование различных метрик [15, 16], нами использована евклидова норма. Для оценки точности взята тестовая выборка базы MNIST, определены пересечения кластеров и вычислены F-меры для каждой цифры.

1. Набор данных MNIST

База данных *MNIST*^{URL} и ее расширения широко используются для тестирования различных подходов к распознаванию образов [17, 18], кластеризации [19], а также других алгоритмов [20–22]. Многие алгоритмы проверяются на этой базе данных. Например, при распознавании образов методы k-ближайшего соседа на основе MNIST дают ошибку в 5%, сверточные нейронные сети – менее 1%, а многослойный персептрон – около 2–5% в зависимости от методов обучения и количества слоев.

Набор данных MNIST – это база данных рукописных цифр, которая содержит 60 000 изображений цифр в обучающей выборке и 10 000 изображений в тестовой. В таблице 1 содержится информация о количестве изображений каждой цифры в обучающей и тестовой выборках.

Таблица 1. Количество изображений каждой цифры в обучающих и тестовых выборках из набора данных MNIST.

Цифра	Обучающая выборка	Тестовая выборка
0	5923	980
1	6742	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009

Каждое изображение из этого набора данных представлено в двух формах: в виде метки и в виде вектора значений пикселей, а также нормировано по размеру. Размер каждого изображения составляет 28 на 28 пикселей. Значение пикселя изменяется от 0 до 255, где 0 соответствует черному пикселю, а 255 – белому.

2. Нейронная сеть Кохонена

Сеть Кохонена представляет собой специальный тип нейронной сети для решения задачи кластеризации. Она использует неконтролируемый тип обучения «без учителя» и состоит из одного слоя настраиваемых весов. Веса нейронной сети изменяются таким образом, что векторы, принадлежащие одному и тому же кластеру, приводят в движение один и тот же выходной нейрон [23]. Архитектура нейронной сети Кохонена приведена на рисунке 1.

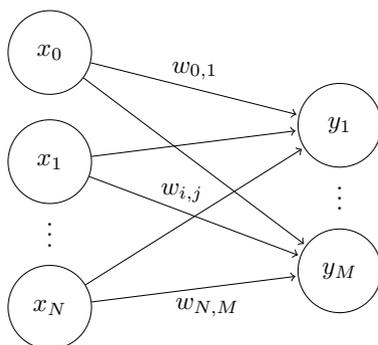


Рисунок 1. Архитектура нейронной сети Кохонена.

Выходы z_j нейронной сети Кохонена вычисляются по формуле

$$z_j = \sum_{i=1}^{N=784} w_{ij}x_i,$$

где x_i – входы нейронной сети Кохонена. Каждое изображения в наборе данных MNIST имеет размер 28×28 , это означает, что количество нейронов в сети Кохонена будет равно $28 \times 28 = 784$. Веса w_{ij} являются центрами кластеров и участвуют в обучении. Ниже представлен алгоритм обучения нейронной сети Кохонена:

1. Входные векторы x^p нормируются.
2. Значения весов w_{ij} выбираются случайным образом из векторов обучающей выборки. Это связано с тем, что в случае неравномерного распределения (если веса будут заполняться случайным образом) веса могут располагаться далеко от входных векторов и по этой причине не принимать участия в обучении.
3. Вычисляются расстояния между входными векторами x^p с координатами x_i^p и весами w_j по следующей формуле:

$$D_j = \sum_{i=1}^{N=784} ((x_i^p - w_{ij})^2)^{1/2}.$$

Выбирается вектор-победитель – такой вектор w_j , у которого расстояние D_j до входного вектора x^p наименьшее.

4. Изменяются координаты вектора победителя, выбранного на предыдущем шаге, в соответствии с приведенной ниже формулой, где θ – скорость обучения:

$$w_l = w_l + \theta(x^p - w_l).$$

5. Шаги 3 и 4 повторяются для всех векторов x^p .

Значения θ на каждой итерации определяются по формуле $\theta = \alpha\theta$, где $\alpha < 1$. Если $\theta > \varepsilon$, то выполняется переход к шагу 3, и снова после прохода через все входные векторы веса корректируются.

3. Число кластеров каждой рукописной цифры

Основная проблема при кластеризации рукописных цифр заключается в том, что все люди имеют разный почерк, значит, одна и та же цифра может быть написано разными способами. На примере набора данных MNIST можно видеть, что существует множество вариантов записи одной и той же цифры. Цифры имеют свои специфические особенности написания, например, цифру 1 можно записать с наклоном или без наклона. Каждая цифра имеет разное количество таких признаков. По этой причине количество кластеров, на которое необходимо разбить множество изображений для каждой цифры, может быть разным. Количество кластеров, на которые необходимо разделить входные изображения, неизвестно. Необходимо определить оптимальное число кластеров для каждой цифры. Для автоматической кластеризации будем использовать следующий алгоритм [24]:

1. На основе евклидовой метрики вычислим квадраты расстояний между всеми векторами в обучающей выборке. Из полученных значений вычислим элементы матрицы расстояний

$$d^2(x^i, x^j) = \|x^i - x^j\|^2 = \sum_{i=1}^N (x_i^i - x_i^j)^2.$$

2. Определим максимальное значение элементов матрицы, построенной на предыдущем шаге. Это значение является максимальным расстоянием между всеми векторами: $\max(d^2(x^i, x^j))$.
3. Определим допустимое расстояние между векторами, расположенными в одном кластере, – оно определяется как фиксированный процент от максимального расстояния между векторами.
4. Выберем произвольный i -й столбец матрицы (вектор x^i). Отметим все элементы столбца со значениями, меньшими допустимых значений, как элементы одного и того же кластера (строки с номером j). При этом проигнорируем i -й столбец, а также все j -й столбцы;
5. Если матрица еще не пуста, то переходим к шагу 4.

На рисунке 2 показано, как изменяется оптимальное количество кластеров в зависимости от заданного процента, описанного на шаге 3. Построим такую зависимость и проверим для каждой цифры.

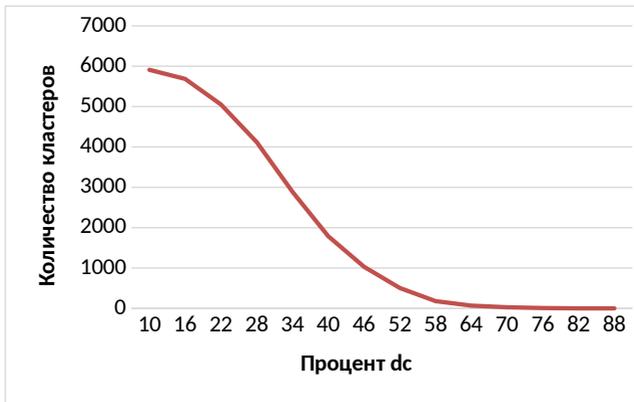


РИСУНОК 2. Зависимость оптимального количества кластеров от заданного процента максимального расстояния.

Анализ полученных зависимостей показал, что они одинаковы для всех цифр. Очевидно, что чем меньше процент максимального расстояния между векторами, тем на большее число кластеров необходимо разделить выборку. Исследуем, сколько кластеров нужно выбрать, при условии, что их количество не превысит 50. Фиксированный процент для каждого показателя выберем отдельно в интервале от 65% до 70%. В результате получим количества кластеров для каждой цифры, которые показано в таблице 2.

Таблица 2. Оптимальные количества кластеров для каждой цифры.

Цифра	Количество кластеров
0	39
1	49
2	30
3	47
4	28
5	33
6	49
7	29
8	38
9	28

После определения оптимальных количеств кластеров для каждой цифры используем нейронную сеть Кохонена для нахождения самих кластеров.

4. Кластеризация нейронной сетью Кохонена

Проведем кластеризацию шестидесяти тысяч изображений цифр нейронной сетью Кохонена. Оптимальные параметры для обучения сети получены экспериментально и имеют следующие значения: $\alpha = 0,96$, $\varepsilon = 0,02$, $\theta = 0,6$.

Проверим правильность построения кластеров на тестовом наборе данных. Для этого на основе тестовой выборки MNIST проанализируем количество векторов-изображений, которые попали в «свой» кластер. Определим расстояние от каждого вектора изображения цифры

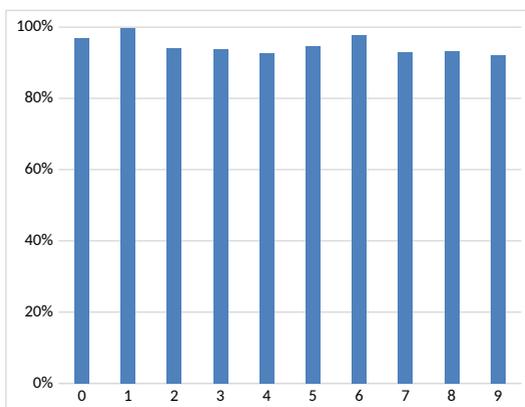


Рисунок 3. Процент изображений цифр, которые попадают из тестовой выборки в свой кластер.

до всех полученных кластеров для каждой из цифр. Если вектор изображения находится ближе всего к кластеру, который относится к группе кластеров для того же рисунка, то считаем, что вектор попал в «свой» кластер. На рисунке 3 представлены процентные доли попадания изображений из тестовой выборки в кластеры их цифр.

Кроме того, для анализа точности результатов кластеризации нейронной сетью Кохонена вычислим F-меру (см. Таблицу 3).

Таблица 3. F-мера для кластеров.

Цифра	Recall	Precision	F-means
0	0,97	0,97	0,97
1	0,96	0,99	0,98
2	0,97	0,94	0,95
3	0,96	0,94	0,95
4	0,94	0,93	0,93
5	0,94	0,96	0,95
6	0,97	0,98	0,97
7	0,95	0,93	0,94
8	0,95	0,93	0,94
9	0,89	0,92	0,91

5. Пересечения кластеров

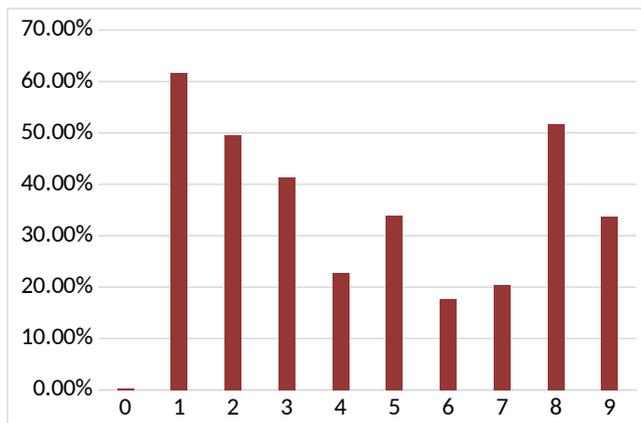
Теперь проанализируем пересечение кластеров друг с другом. Исследовав пересечения кластеров, можно ответить на вопрос, подходит ли кластеризация с помощью нейронной сети Кохонена для распознавания паттернов. Очевидно, что чем меньше будет пересечений кластеров между собой, тем точнее распознавание.

Основная проблема распознавания (и кластеризации) образов заключается в том, что существует «бесконечное множество» способов написать одну и ту же цифру. Это приводит к тому, что либо количество кластеров значительно увеличивается, либо увеличивается размер самих кластеров. Также существует проблема: разные цифры записываются похожим или даже почти одинаковым образом. Значит, в случае небольшого количества кластеров можно ожидать, что кластеры из-за своего большого размера будут иметь множественные пересечения с кластерами других цифр.

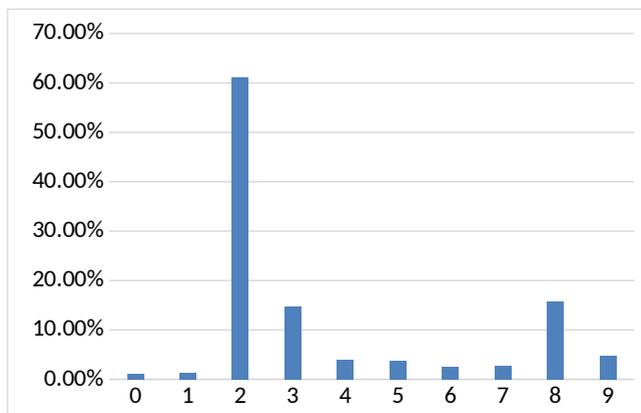
Чтобы найти пересечения между кластерами, необходимо определить, на каком расстоянии от центра каждого кластера находится цифра из выборки. Если расстояние до цифры меньше радиуса кластера, то считаем, что цифра находится в этом кластере. Исследуем пересечения полученных кластеров с изображениями из тестовой выборки.

- Кластеры для цифры 0 пересекаются с кластерами для цифр 2 и 5. Это связано с тем, что эти цифры имеют одинаковые «закругления».
- Кластеры для цифры 1 имеют большее количество пересечений, чем другие кластеры. Это связано с тем, что элемент цифры 1 содержится в изображении других цифр.
- Кластеры для цифры 2 имеют значительные пересечения с кластерами для цифр 3 и 8.
- Кластеры для цифры 3 имеют пересечения с цифрами 2, 5 и 8.
- Кластеры для цифр 4 и 9 имеют много пересечений между собой.
- Кластеры для цифры 5 пересекаются с кластерами для цифр 3 и 8.
- Кластеры для цифры 6 пересекаются с кластерами для цифры 2.
- Кластеры для цифры 7 пересекаются с кластерами для цифры 9.
- Кластеры для цифры 8 пересекаются с кластерами для цифр 2 и 9.

На рисунке 4 приведены примеры диаграмм, показывающих, какой процент изображений каждой цифры из тестовой выборки принадлежит кластеру рассмотренной цифры.



(а) для цифры 1



(б) для цифры 2

Рисунок 4. Примеры пересечений кластеров для цифр

Например, на рисунке 4(а) кирпичный цвет (верхний правый сектор) показывает, что 61,64% изображений цифры 1 принадлежат, в дополнение к их собственному кластеру цифры 1, к другому кластеру

цифры 1. Фактически это означает, что 61,64% изображений цифры 1 принадлежат двум и более кластерам цифры 1. Если переместиться по часовой стрелке в следующий (серый) сектор для цифры 2, то видим, что 49,57% изображений двойки попадают, помимо своего кластера, в кластер цифры 1.

Также обратим внимание (это не отображено на диаграммах), что многие изображения цифры 1 попадают в кластеры разных цифр. Анализ результатов пересечения показал, что из тестовой выборки цифры 1 много изображений попало в пересечение с изображениями других цифр: почти 23% тестовой выборки цифры 1 попали в пересечение кластеров для цифры 0; почти 50% – с кластерами для 2 и так далее. Это связано с тем, что элементы изображения других цифр содержат изображение цифры 1.

Для цифры 2 ситуация иная (рисунок 4(б)). Существуют одно большое пересечение с кластерами для цифры 2 (самопересечения характерны для кластеров всех цифр) и небольшие пересечения с кластерами других цифр.

Приведем примеры изображений, которые попали в пересечения своего кластера и кластера другой цифры (рисунок 5).



(а) 0 в пересечении с 6 (б) 1 в пересечении с 6 (в) 5 в пересечении с 3

Рисунок 5. Примеры цифр, которые расположены в пересечениях кластеров:

В этих примерах нейронная сеть правильно группирует изображения. На рисунке 5(а) цифра была распознана как 0, но попала в пересечение с кластером для цифры 6. На рисунке 5(б) цифра распознана как 1, но принадлежит также кластеру для цифры 6, на рисунке 5(в) цифра распознана как цифра 5, но принадлежит также кластеру для цифры 3.

Обсуждения

Основная проблема при кластеризации изображений цифр заключается в том, что одни и те же цифры записываются по-разному. Это приводит к увеличению размеров отдельных кластеров или их количества. Также существует проблема, связанная с тем, что разные цифры записываются похожим образом. Поэтому, в случае небольшого количества кластеров, ожидается, что из-за их большого размера будут пересечения кластеров между собой.

Проанализировав рисунок 4 и таблицу 3, можно сделать вывод, что нейронная сеть Кохонена может быть использована для кластеризации и распознавания рукописных цифр, поскольку большой процент изображений цифр из тестовой выборки попадает в требуемый кластер. Отметим, что цифра 9 признана наихудшей – процент распознавания чуть более 90%. Цифра 1 распознается нейронной сетью Кохонена почти в 100% случаев.

Другие нейронные сети также могут быть использованы для решения задач распознавания [25, 26]. Выводы о принадлежности изображения своему кластеру могут быть использованы в качестве одного из этапов распознавания. Например, кластеризация может быть использована в качестве такого этапа в иерархической нейронной сети [27, 28].

Стоит отметить, что предложенный метод кластеризации для ограниченного числа кластеров может быть использован в задаче распознавания нейронной сетью Хопфилда, так как у этой Хопфилда существует ограничение на количество запоминаемых объектов (кластеров).

Заключение

Проведен кластерный анализ рукописных цифр из базы данных MNIST с использованием нейронной сети Кохонена. Для каждой цифры определено оптимальное количество кластеров. Изображения из тестовой выборки принадлежат правильному кластеру с вероятностью более 90% для каждой цифры.

Сделан вывод, что кластеризация может быть использована для распознавания рукописных цифр. Наилучшая кластеризация получена для цифр 0 и 1 (F-мера равна 0,97). Наихудшая кластеризация получена для цифры 9 – F-мера равна 0,903. Определены пересечения кластеров и проанализировано, какие цифры с какими кластерами пересекаются. Приведены примеры отдельных характерных изображений рукописных цифр, которые попали в пересечения кластеров.

Предложенный подход может быть использован для кластеризации больших объемов данных различного типа и различной сложности.

Список литературы

- [1] Latypova D. *Neural networks using for handwritten numbers recognition*, Master's Thesis.– Prague: Czech Technical University.– 2020.– 77 pp. [URL](#) ↑226
- [2] Remy M., Lavanya K. *Handwritten digit recognition of MNIST data using consensus clustering* // International J. of Recent Technology and Engineering.– 2019.– Vol. 7.– No. 6.– pp. 1969–1973. [URL](#) ↑226
- [3] Nhery S., Ksantini R., Kaaniche M. B., Bouhoula A. *A novel handwritten digits recognition method based on subclass low variances guided support vector machine* // Proc. of the 13th Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Application.– V. 4, VISIGRAPP 2018 (27–29 January 2018, Funchal, Madeira, Portugal).– 2018.– ISBN 978-989-758-290-5.– pp. 28–36. [URL](#) ↑226
- [4] Shal S. A., Koltun V. *Robust continuous clustering* // Proc. the Natl. Acad. Sci. USA.– 2017.– Vol. 114.– No. 37.– pp. 9814–9817. [doi](#) ↑226
- [5] Miri E., Razavi S. M., Sadri J. *Performance optimization of neural networks in handwritten digit recognition using Intelligent Fuzzy C-Means clustering*, 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE) (13–14 Oct. 2011, Mashhad, Iran).– 2011.– pp. 150–155. [doi](#) ↑226
- [6] Pourmohammad S., Soosahabi R., Maida A. S. *An efficient character recognition scheme based on k-means clustering*, 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO) (28–30 April 2013, Hammamet, Tunisia).– 2013.– pp. 1–6. [doi](#) ↑226
- [7] Li B. Y. *An experiment of k-means initialization strategies on handwritten digits dataset* // Intelligent Information Management.– 2018.– Vol. 10.– No. 2.– pp. 43–48. [doi](#) ↑226
- [8] Munggaran L. C., Widodo S., Cipta A. M., Nuryuliani *Handwritten pattern recognition using Kohonen neural network based on pixel character* // International J. of Advanced Computer Science and Applications.– 2014.– Vol. 5.– No. 11.– 6 pp. [doi](#) [URL](#) ↑226
- [9] Fahad A., Alshatri N., Tari Z., Alamri A., Khalil I., Zomaya A., Fofou S., Bouras A. *A survey of clustering algorithms for big data: taxonomy and empirical analysis* // IEEE Transactions on Emerging Topics in Computing.– 2014.– Vol. 2.– No. 3.– pp. 267–279. [doi](#) ↑226
- [10] Bi Y., Wang P., Guo X., Wang Z., Cheng S. *K-means clustering optimizing deep stacked sparse autoencoder* // Sensing and Imaging.– 2019.– Vol. 20.– No. 1.– 6.– 19 pp. [doi](#) ↑226
- [11] Chen Y., Chen C.-G., You C. *Stochastic sparse subspace clustering*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (13–19 June 2020, Seattle, WA, USA).– 2020.– pp. 4155–4164. [doi](#) ↑226
- [12] Zhang S., You C., Vida R., Li C. G. *Learning a self-expressive network for subspace clustering*, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (20–25 June 2021, Nashville, TN, USA).– 2021.– pp. 12393–12404. [doi](#) ↑226

- [13] Латыпова Д., Тумаков Д. *Определение основных кластеров рукописных цифр*, Цифровая обработка сигналов и ее применение. DSPA - 2020 (14–15 апреля 2020 года, Москва, Россия).– 2020.– с. 620–625. ✨↑226
- [14] Latypova D., Tumakov D. *Peculiarities of image recognition by the Hopfield neural network*, IEMAICLOUD 2021: International Conference on Intelligent Emerging Methods of Artificial Intelligence & Cloud Computing, Smart Innovation, Systems and Technologies.– vol. **273**, ed. Garcia Marquez F.P., Cham: Springer.– 2021.– ISBN 978-3-030-92904-6.– pp. 34–47. doi↑227
- [15] McConnell S., Sturgeon R., Henry G., Mayne A., Hurley R. *Scalability of self-organizing maps on a GPU cluster using OpenCL and CUDA* // High Performance Computer Symposium, J. Physics: Conf. Series.– 2012.– Vol. **341**.– 012018.– 10 pp. doi↑227
- [16] Xu Y., Zhang W. *On a clustering method for handwritten digit recognition*, 2010 Third International Conference on Intelligent Networks and Intelligent Systems (01–03 November 2010, Shenyang, China).– 2010.– pp. 112–115. doi↑227
- [17] Cohen G., Afshar S., Tapson J., van Schaik A. *EMNIST: extending MNIST to handwritten letters*, 2017 International Joint Conference on Neural Networks (IJCNN) (14–19 May 2017, Anchorage, AK, USA).– 2017.– pp. 2921–2926. doi↑227
- [18] Baldominos A., Saez Y., Isasi P. *A survey of handwritten character recognition with MNIST and EMNIST* // J. of Applied Science.– 2019.– Vol. **9**.– No. 15.– 3169.– 16 pp. doi↑227
- [19] Agarap A. F., Azcarraga A. P. *Improving k-means clustering performance with disentangled internal representations*, 2020 International Joint Conference on Neural Networks (IJCNN) (19–24 July 2020, Glasgow, UK).– 2020.– pp. 1–8. doi↑227
- [20] Cheng K., Tahir R., Eric L. C., Li M. *An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset* // J. Multimedia Tools and Applications.– 2020.– Vol. **79**.– pp. 13725–13752. doi↑227
- [21] Kossen J., Farquhar S., Gal Y., Rainforth T. *Active testing: sample-efficient model evaluation* // *Proc. of the 38th International Conf. on Machine Learning* (18–24 July 2021, Virtual), Proceedings of Machine Learning Research.– vol. **139**.– 2021.– pp. 5753–5763. URL↑227
- [22] Zhang R., Chang P.-C. *Robustness against adversary models on MNIST by deep-Q reinforcement learning based parallel-GANs*, 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (14–17 December 2021, Tokyo, Japan).– 2021.– pp. 1590–1597. URL↑227
- [23] Murtagh F., Hernandez-Pajares M. *The Kohonen self-organizing map method: An assessment* // J. of Classification.– 1995.– Vol. **12**.– pp. 165–190. doi↑228
- [24] Сеньковская И. С., Сараев П. В. *Автоматическая кластеризация в анализе данных на основе саморганизующихся карт Кохонена* // Вестник МГТУ им. Г. И. Носова.– 2011.– № 2(34).– с. 78–79. ✨URL↑229
- [25] Simard P. Y., Steinkraus D., Platt J. C. *Best practices for convolutional neural networks applied to visual document analysis* // *Proc. Seventh International*

- Conference on Document Analysis and Recognition* (06 August 2003, Edinburgh, UK).– 2003.– ISBN 0-7695-1960-1.– pp. 958–963. doi ↑236
- [26] Ciresan D., Meier U., Schmidhuber J. *Multi-column deep neural networks for image classification*, 2012 IEEE Conference on Computer Vision and Pattern Recognition (16–21 June 2012, Providence, RI, USA).– 2012.– pp. 3642–3649. doi ↑236
- [27] Kayumov Z., Tumakov D., Mosin S. *Hierarchical convolutional neural network for handwritten digits recognition* // Proc. Computer Science.– 2020.– Vol. 171.– pp. 1927–1934. doi ↑236
- [28] Kayumov Z., Tumakov D., Mosin S. *Combined convolutional and perceptron neural networks for handwritten digits recognition*, 2020 22th International Conference on Digital Signal Processing and its Applications (DSPA) (25–27 March 2020, Moscow, Russia).– 2020.– pp. 74.– 5 pp. doi ↑236

Поступила в редакцию 28.06.2022;
одобрена после рецензирования 12.08.2022;
принята к публикации 10.09.2022.

Рекомендовал к публикации

д.ф.-м.н. А. М. Елизаров

Информация об авторах:



Дина Сергеевна Латыпова

аспирант Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета. Область научных интересов – искусственный интеллект, нейронные сети, высокопроизводительные вычисления.

 0000-0002-3282-8545
e-mail: dina.latypova23@gmail.com



Дмитрий Николаевич Тумаков

к. ф.-м. н., заместитель директора Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета по научной деятельности. Область научных интересов – машинное обучение, искусственный интеллект, машинное зрение, распознавание образов, математическое моделирование, высокопроизводительные вычисления.

 0000-0003-0564-8335
e-mail: dtumakov@kpfu.ru

*Все авторы сделали эквивалентный вклад в подготовку публикации.
Авторы заявляют об отсутствии конфликта интересов.*