

УДК DOI 10.31891/2307-5732-2024-339-4-61
УДК 004.85

ОДЕГОВ МИКОЛА

Державний університет інтелектуальних технологій і зв'язку
<https://orcid.org/0000-0001-5526-2487>
e-mail: onick_64@ukr.net

БАГАЧУК ДЕНИС

Державний університет інтелектуальних технологій і зв'язку
<https://orcid.org/0000-0001-8798-891X>
e-mail: bagachukdg@gmail.com

ПЕРЕКРЕСТОВ ІГОР

Державний університет інтелектуальних технологій і зв'язку
<https://orcid.org/0009-0007-3805-8143>
e-mail: perekrestov.igor@gmail.com

ПЕТРОВИЧ ЯННА

Державний університет інтелектуальних технологій і зв'язку
e-mail: yanna-petrovich@ukr.net

ПРИНЦИП ДАЛЬНОЇ І БЛИЗЬКОЇ ДІЇ В ЗАДАЧАХ СТРУКТУРИЗАЦІЇ ТА НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

У даній роботі виконується порівняльний аналіз трьох алгоритмів кластеризації за критеріями точності та швидкості. Базовим алгоритмом для порівняння обрано відомий алгоритм *k-means*, який за визначеними критеріями порівнюється з алгоритмом штучних нейронних мереж, які побудовані за принципом далекодії-близькодії (ПДБ-мережі) та гібридним алгоритмом, котрий включає як операції ПДБ-мереж, так і операції *k-means*.

Основною ідеєю ПДБ-мереж є заміна гіперпараметрів на параметри мережі. А саме, замість навчання великої кількості елементів перехідних матриць зв'язків між сусідніми шарами мережі навчається значно менша кількість нейронів, точніше – функцій активації нейронів. Такі функції активації мають мати не менше одного параметру, що підлягає навчанню. У даній роботі в якості таких функцій пропонується використовувати параметричний сигмоїд з одним лише вільним параметром – коефіцієнтом нелінійності.

Гібридний алгоритм передбачає використання ПДБ-мережі для нелінійної проєкції первинних зразків із фактор-простору великої розмірності у фактор-простір з меншою кількістю координат. Далі в новому, стислому, просторі задача кластеризації вирішується за допомогою відомого алгоритму *k-means*.

Алгоритми порівнювались на прикладі 50000 зразків із відомого набору рукописних цифр MNIST. Кількість кластерів в усіх випадках була фіксованою: 1000 кластерів.

В результаті випробувань алгоритмів при різних значеннях гіперпараметрів встановлено, що алгоритми ПДБ-мереж завжди перевищують інші алгоритми за критерієм швидкості та у найкращих варіантах дають дуже прийнятні результати за критеріями точності. Втім, найкращі результати показав гібридний алгоритм, який не дуже сильно відстає від ПДБ-мереж за показником швидкості, але переважає всі інші алгоритми за критеріями точності. Так у найкращому варіанті результат валідації цього алгоритму дає 98,8% вірних рішень задачі кластеризації.

Ключові слова: штучні нейронні мережі, принцип далекодії-близькодії, навчання без вчителя, нелінійні згорткові мережі, параметричний сигмоїд, алгоритм *k-means*.

ODEGOV NICK, BAHACHUK DENYS, PEREKRESTOV IHOR, PETROVYCH YANNA
State University of Intellectual Technologies and Telecommunications

PRINCIPLE OF LONG-AND-CLOSE-RANGE ACTION IN STRUCTURIZATION PROBLEMS AND TRAINING OF ARTIFICIAL NEURAL NETWORKS

In this work, a comparative analysis of three clustering algorithms is performed according to the criteria of accuracy and speed. The well-known *k-means* algorithm was chosen as the basic algorithm for comparison, which, according to certain criteria, is compared with the algorithm of artificial neural networks, which are built according to the principle of long-range-near-range (LRNR-networks) and a hybrid algorithm that includes operations of LRNR-networks, as well as operations of *k-means*.

The main idea of LRNR networks is to replace hyperparameters with network parameters. Namely, instead of training a large number of elements of transition matrices of connections between neighboring layers of the network, a much smaller number of neurons, or more precisely, the activation functions of neurons, are trained. Such activation functions must have at least one learnable parameter. In this work, it is proposed to use a parametric sigmoid with only one free parameter - the nonlinearity coefficient - as such functions.

The hybrid algorithm involves the use of a LRNR network for the nonlinear projection of primary samples from a large-dimensional factor space into a factor space with a smaller number of coordinates. Next, in a new, compressed space, the problem of clustering is solved using the well-known *k-means* algorithm.

Algorithms were compared on the example of 50,000 samples from the well-known MNIST set of handwritten digits. The number of clusters in all cases was fixed: 1000 clusters.

As a result of testing the algorithms with different values of hyperparameters, it was established that the LRNR-network algorithms always exceed other algorithms according to the speed criterion and, in the best cases, give very acceptable results according to the accuracy criteria. However, the best results were shown by the hybrid algorithm, which is not far behind LRNR networks in terms of speed, but outperforms all other algorithms in terms of accuracy. Thus, in the best case, the result of the validation of this algorithm gives 98.8% correct solutions of the clustering problem.

Keywords: artificial neural networks, long-range-near-range principle, learning without a teacher, nonlinear convolutional networks, parametric sigmoid, *k-means* algorithm.

Аналіз джерел та постановка проблеми

Дана робота продовжує серію статей кафедри ІПЗ ДУІТЗ [1-4] присвячених дослідженням у напрямку розробки алгоритмів для вирішення задач класу Big Data. В даній роботі досліджуються різні алгоритми навчання без вчителя, конкретно – алгоритми кластеризації.

Сучасні тенденції вирішення задач кластеризації полягають, в основному, у використанні згорткових нейронних мереж [5, 6]. При цьому значна кількість публікацій присвячена кластеризації зображень (наприклад, [7, 8]). У цитованих роботах спільним є використання методу k-середніх, або k-центроїдів (k-means) на кінцевому етапі кластеризації.

Узагальнений підхід на основі згорткових мереж виявляється досить продуктивним, але він має певні та відомі недоліки. А саме: операції згортки залежать від способу розбиття матриць зображень на підматриці; навчання внутрішніх шарів мереж потребує достатньо багатого часу.

Заслуговує уваги також розробка алгоритмів, які застосовують ті чи інші робастні методи обробки даних в умовах великих шумів, наприклад, дослідження у роботі [9]. Також досить новим напрямком є алгоритми кластеризації із використанням графових структур [10].

Зауважимо, що практично всі алгоритми кластеризації тим чи іншим чином або використовують методу k-means [7, 8], або використовують алгоритми k-means для порівняльного аналізу [10].

У даній роботі алгоритм k-means також використовується як своєрідний еталон кластерного аналізу. При цьому застосовуються класи та методи бібліотеки scikit-learn [11] мови Python.

Метою роботи є порівняння за критеріями швидкості та точності трьох алгоритмів:

- класичного алгоритму k-means;
- алгоритму штучних нейронних мереж, заснованих на принципі далекодії-близькодії (ПДБ-мереж) [4];
- гібридного алгоритму, який застосовує як ПДБ-мережі, так і алгоритми k-means.

Методика порівняльного аналізу алгоритмів

Порівняльний аналіз у даній роботі виконується лише на окремому, але на досить важкому прикладі – вирішення задачі кластеризації на множині рукописних цифр MNIST. Точніше, оскільки вирішується задача навчання без вчителя, то використовуються лише зразки із навчального набору даних в кількості 50000.

Формальна задача: розподілити ці зразки між 1000 кластерами невідомої структури. Головне, щоб вони були різними. Кількість кластерів задається фіксованою.

Всі алгоритми, які порівнюються, мають спільні компоненти, але можуть відрізнятися параметрично:

- вивчення та аналіз предметної сутності задачі (спільне для всіх алгоритмів);
- завантаження даних в оперативну пам'ять та первинна обробка цих даних (практично, спільне для всіх алгоритмів);
- визначення гіперпараметрів алгоритмів (різне для різних алгоритмів);
- навчання алгоритмів (різне для різних алгоритмів);
- валідація результатів кластеризації (спільне для всіх алгоритмів).

Зауважимо, що всі дослідження виконувались лише за допомогою CPU звичайного комп'ютера (швидкісні алгоритми із застосуванням GPU не використовувались).

Предметна область – рукописні цифри з тестового набору MNIST, приклади яких показано на рис. 1. Набір MNIST є розміченим набором даних. Тобто, поряд з масивом образів зображень 28x28 пікселів (images) є відповідний масив міток образів (labels). В усіх алгоритмах для вирішення задачі кластеризації масив labels, зрозуміло, не використовується.

Попередній аналіз зображень рукописних цифр показує, що задача кластеризації, рівно як і задача класифікації, на даному наборі зображень є досить складним тестом для відповідних алгоритмів. Дійсно, якщо уважно роздивитись приклади рукописних цифр на рис. 1, то при аналізі зліва направо можна відмітити дивні картини.

Так, у другій позиції цифри 2 взагалі незрозуміле для людського ока зображення, а у десятій позиції двійка скоріше нагадує сімку. Образи, які розмічені як 6 також можуть дуже сильно відрізнятися від шістки. Так у сьомій та дев'ятій позиціях шістку важко відрізнити від нуля. Сімка у сьомій позиції дуже нагадує одиницю.

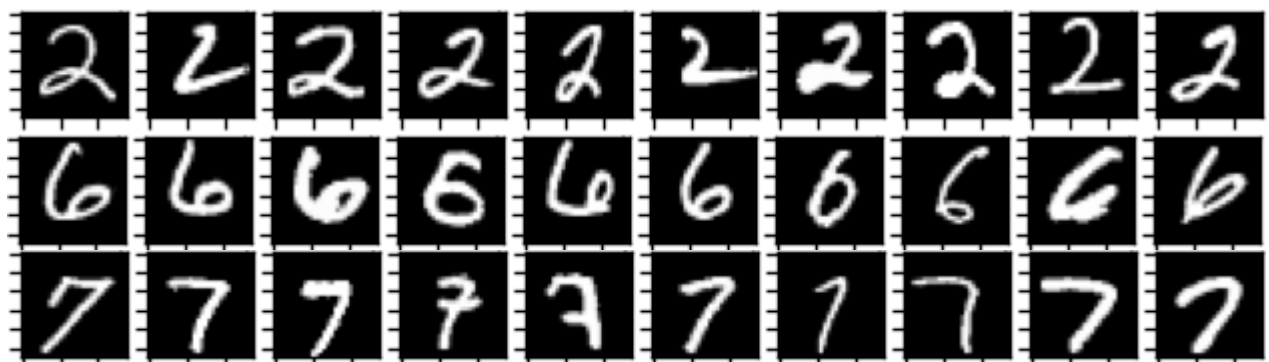


Рис. 1. Приклади зображень рукописних цифр набору MNIST

Таких прикладів, коли образи з певними мітками маскуються під інші цифри можна навести безліч. Тому повторюємо: задача кластеризації даних образів є дуже суворим тестом для відповідних алгоритмів.

На попередньому етапі після завантаження бітових потоків із бази даних MNIST, фрагменти цих потоків 28x28 байтів перетворюються у масиви розмірності 784 елементів, які і обробляються далі алгоритмами кластеризації.

Для валідації суттєво використовуються мітки (labels). При цьому для кожного з 1000 кластерів визначається медіана міток образів, що потрапили у даний кластер. Значення медіан, зрозуміло, буде на множині 0, 1, 2, ..., 9. Кількість зразків з однаковими медіанами підсумовуються і порівнюються з відповідною кількістю зразків у класах MNIST. Для аналізу важливо, за якими показниками визначається точність алгоритмів. У даній роботі використовуються два показники: середнє відносне відхилення D_{mean} та максимальне відносне відхилення D_{max} , які визначаються очевидним образом:

$$D_{mean} = \frac{1}{10} \frac{\sum_{m=0}^9 |N_{MNIST}(m) - N_{clust}(m)|}{\sum_{m=0}^9 N_{MNIST}(m)}; \quad D_{max} = \frac{\max_m |N_{MNIST}(m) - N_{clust}(m)|}{\sum_{m=0}^9 N_{MNIST}(m)}, \quad (1)$$

де m – номер цифри (0, 1, 2, ..., 9); $N_{MNIST}(m)$ – сумарна кількість зразків цифри m у множині даних MNIST; $N_{clust}(m)$ – сумарна кількість зразків цифри m , отриманих в результаті кластеризації.

Результати кластеризації, отримані за допомогою алгоритму k-means

Використання алгоритмів k-means за допомогою швидкісних засобів бібліотеки scikit-learn у простому варіанті потребує визначення таких гіперпараметрів:

- Method: метод визначення початкових позицій центроїдів кластерів (випадково або за критерієм найбільшої відстані між первинним положенням центроїдів);
- n_init: кількість циклів для первинного визначення положень центроїдів;
- max_iter: кількість ітерацій для уточнення положень центроїдів і віднесення зразків до кластерів.

Ще один гіперпараметр визначено вище: загальна кількість кластерів задана жорстко і дорівнює 1000. У даній роботі також однозначно обрано параметр Method, який при ініціалізації об'єкту KMeans задається як `init='k-means++'`. Тобто, для ініціалізації центроїдів обираються найбільш видалені зразки зображень.

На рис. 2 та 3 графічно представлені деякі варіанти валідації кластерного аналізу з використанням алгоритму k-means при різних значеннях гіперпараметрів `n_init` та `max_iter`.

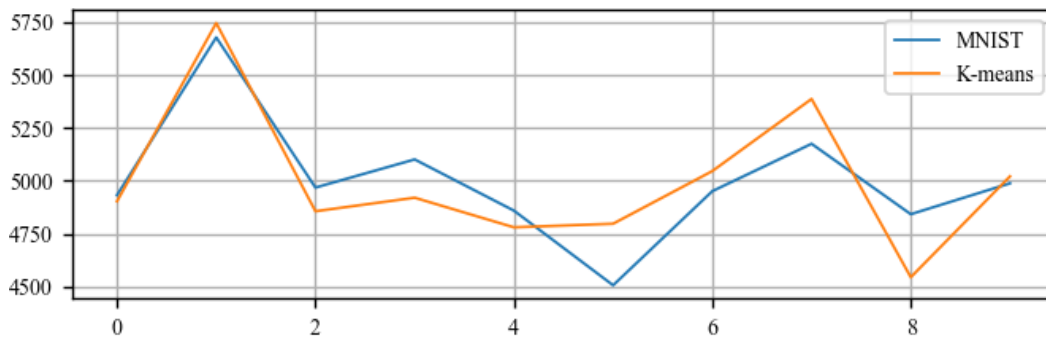


Рис. 2. Результати валідації при `n_init = 2`, `max_iter = 10`

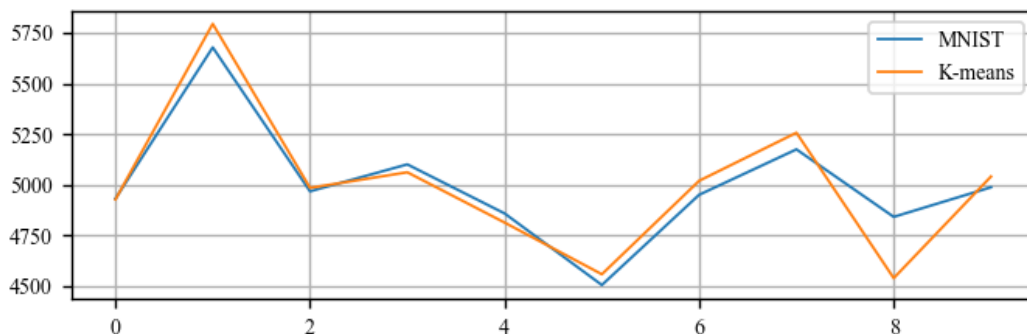


Рис. 3. Результати валідації при `n_init = 30`, `max_iter = 20`

Аналіз графіків на рис. 2 та 3 показує, що при збільшенні кількості ітерацій дещо збільшується і точність алгоритмів. Втім, як і при малій, так і при великій кількості ітерацій алгоритм k-means дає найбільші похибки при розпізнаванні образів цифри 8.

Більш докладно результати тестування алгоритму k-means наведені у табл. 1

Таблиця 1

Результати порівняльного аналізу результатів кластеризації за допомогою алгоритму k-means

Гіперпараметри			Параметри рішень		
Method	n_init	max_iter	Dmean %	Dmax %	Час, с
k-means++	2	10	2.796	5.962	607
k-means++	10	2	2.128	6.584	962
k-means++	20	10	2.012	4.859	1739
k-means++	30	20	1.559	6.040	2819

Аналіз даних у табл. 1 показує, що із збільшенням кількості ітерацій суттєво зростає час вирішення задачі. Так, тривалість обробки даних складає приблизно від 10 хвилин до 47 хвилин. При цьому параметри точності покращуються не дуже суттєво. Найкращий варіант за показником відносного середнього відхилення (Dmean % приблизно 1,5 відсотка) отримано за досить довгий час порядку 47 хвилин. Найкращий варіант за показником відносного максимального відхилення (Dmax % приблизно 5 відсотків) отримано за ледь не півгодини.

Втім, з урахуванням складності вхідних образів (рис. 1) всі ці результати навчання без вчителя можна вважати більш-менш прийнятними.

Результати, отримані за допомогою нейронної мережі, заснованою за принципом далекодії-близькодії

Принципи та алгоритми побудови структур нейронних мереж, заснованих за принципом далекодії-близькодії (ПДБ-мережі) докладно наводяться у роботі [4]. У даній роботі нагадаємо лише основні положення.

Основна ідея ПДБ-мереж полягає у тому, що замість навчання перехідних матриць зв'язків між сусідніми шарами навчаються лише функції активації нейронів. Це забезпечує суттєве підвищення швидкості алгоритмів. Так, якщо у сусідніх шарах відносно невелика кількість 1000 нейронів, то у класичних мережах, заснованих на принципах перцептронів, треба навчати матрицю із 1 000 000 елементів. У ПДБ-мережах навчаються лише 1000 нейронів.

Для забезпечення такого підходу перехідні матриці визначаються лише один раз на початку роботи мережі. Параметри цих матриць залежать від задачі, що вирішується. Втім, вони визначаються, умовно кажучи, з якихось «розумних міркувань». У ПДБ-мережах використано дуже просте міркування: чим ближче нейрон наступного шару до нейрону попереднього шару, тим більший вплив надає перший нейрон на другий.

Більш конкретно, умовно нейрони сусідніх шарів розміщуються як би у радіальній структурі, де шари мають свій радіус, а нейрони розміщуються з рівним кроком по куту розміщення[4].

Нейрони шарів при цьому повинні мати не менше одного вільного параметру функції активації, який і підлягає навчанню. Зокрема, у роботі [4] пропонується використовувати параметричний сигмоїд з одним вільним параметром α – коефіцієнтом нелінійності:

$$\sigma(\alpha, x) = \frac{1}{1 + \exp[-\alpha(x - 0,5)]} \quad (2)$$

Як показують попередні дослідження, дуже прийнятні з практичної точки зору рішення можна отримати, якщо оптимізувати відразу для всіх нейронів і всіх шарів ПДБ-мережі лише одне значення коефіцієнту нелінійності. Оптимальне значення Alpha_ort цього коефіцієнту обирається із діапазону значень від Alpha_min до Alpha_max. Даний діапазон суттєво залежить від структури ПДБ-мережі і визначається експериментальним шляхом. Значення Alpha_min та Alpha_max, зрозуміло, відносяться до гіперпараметрів алгоритму.

Найважливіший гіперпараметр при цьому – опис структури мережі, яка задається кількістю нейронів у сусідніх шарах. Це можна показати на прикладі структур, що розглядаються нижче у псевдокоді в стилі мови Python:

```
STRUCTURE1 = ((784, 512), (512, 256), (256, 128), (128, 64))
```

```
STRUCTURE2 = ((784, 512), (512, 256))
```

```
STRUCTURE3 = ((784, 512), (512, 128))
```

Деякі з варіантів визначення оптимального значення коефіцієнту нелінійності та результатів валідації показано на рис. 4 та 5.

Аналіз графіків на рис. 4 та 5 показує, що алгоритми ПДБ-мережі більш рівномірно розподіляють помилки валідації між значеннями міток первинних образів. Наприклад, помилки ідентифікації цифри 8 менш помітні. Також можна побачити, що значення оптимального коефіцієнту нелінійності Alpha_ort суттєво залежить від структури мережі.

Додаткові дослідження дозволили зробити ще один висновок: чим більше шарів у ПДБ-мережі, тим менше значення має коефіцієнт нелінійності. Пояснення цьому ефекту може бути досить простим: перетворення за допомогою функцій активації стають все більше подібні до лінійних, а нелінійні операції в основному виконують дискретні перетворення за допомогою перехідних матриць між сусідніми шарами мережі.

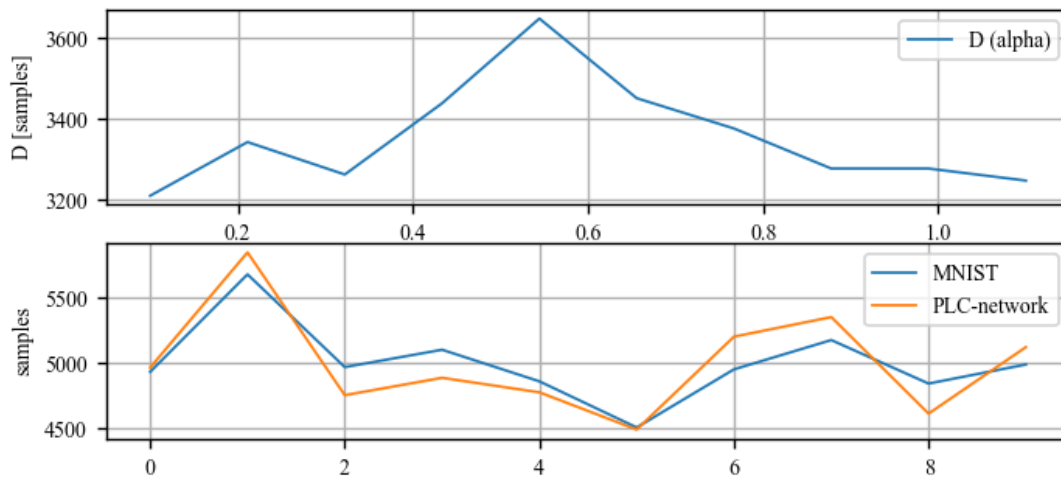


Рис. 4. Результати визначення параметру Alpha_opt (зверху) та результати валідації із застосуванням алгоритму ПДБ-мережі зі структурою STRUCTURE1 (знизу)

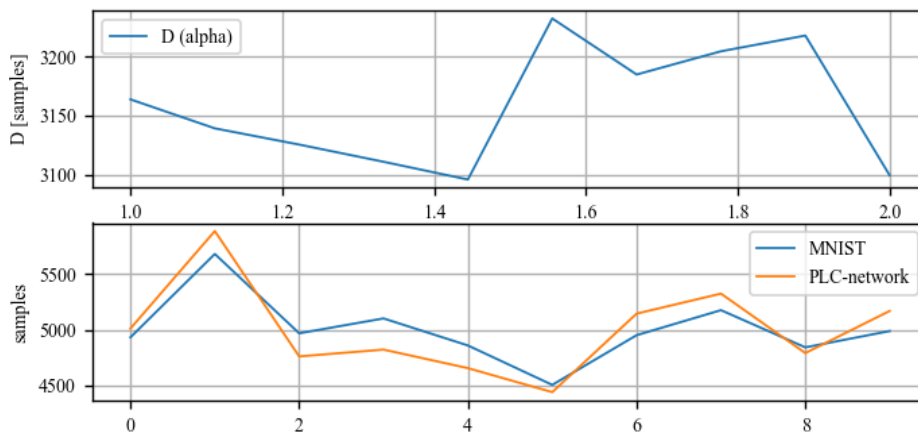


Рис. 5. Результати визначення параметру Alpha_opt (зверху) та результати валідації із застосуванням алгоритму ПДБ-мережі зі структурою STRUCTURE2 (знизу)

Зауважимо, що такі перетворення мають математичну сутність як нелінійні дискретні згортки.

Більш формальні дані для порівняння результатів, отриманих за допомогою алгоритмів ПДБ-мереж, наведені у табл. 2.

Таблиця 2

Результати порівняльного аналізу результатів кластеризації за допомогою ПДБ-мереж

Гіперпараметри			Параметри рішень			
Structure	Alpha_min	Alpha_max	Alpha_opt	Dmean %	Dmax %	Час, с
STRUCTURE1	0.1	1.1	0.544	3.04	4.720	438
STRUCTURE2	1.0	2.0	1.555	3.22	5.579	436
STRUCTURE3	1.0	2.0	1.222	5.51	8.978	430

Аналіз даних у табл. 2 та порівняння з даними у табл. 1 дозволяє зробити наступні висновки:

- збільшення глибини мережі (кількості шарів) дозволяє, як правило, збільшувати точність рішень;
- збільшення кількості шарів незначно впливає на показник швидкості;
- найкращі результати за точністю отримані для структури мережі STRUCTURE1;
- за показником швидкості ПДБ-мережі в будь-якому випадку перевищують алгоритми k-means;
- за показником точності результат ПДБ мережі по показнику Dmax % також є найкращим;
- за показником середньої кількості помилок Dmean % алгоритм ПДБ-мереж все ж таки поступається алгоритму k-means.

Результати, отримані за допомогою гібридного алгоритму

Отримані у попередніх розділах цієї роботи призводять до очевидного висновку доцільності поєднання переваг обох алгоритмів в одному гібридному алгоритмі.

В алгоритмах ПДБ-мереж після навчання мережі (визначення параметру Alpha_opt) первинні центроїди визначаються випадковим чином. Далі віднесення до кластерів відбувається методом найближчого сусіда з використанням якоїсь певної метрики. На кожному кроці положення центроїдів, зрозуміло, корегуються за допомогою простих рекурентних алгоритмів.

Пропонується використання гібридного алгоритму, який складається із наступних послідовних операцій:

- за допомогою алгоритмів навчання ПДБ-мереж з'ясовується оптимальне значення коефіцієнту нелінійності α_{opt} ;
- після цього всі вхідні зразки за допомогою ПДБ-мережі перетворюються у новий набір даних в іншому фактор-просторі;
- у новому фактор-просторі за допомогою швидкого алгоритму k-means визначаються положення центроїдів;
- далі перетворені зразки відносяться до обраних кластерів методом найближчого сусіда із застосуванням тієї чи іншої певної метрики.

Даний алгоритм, таким чином, засновується на дуже швидких операціях ПДБ-мереж. Застосування алгоритму k-means також може бути досить швидким. Справа в тому, що на вході у даному випадку фактор-простір має розмірність 784 координати. Після нелінійної проєкції за допомогою ПДБ-мережі в результаті розмірність фактор-простору може бути значно зменшеною. Так, для наведених вище прикладів структур (STRUCTURE2 та STRUCTURE3) розмірність фактор-просторів зменшується відповідно до 256 та 128 координат. У графічному вигляді результати валідації для цих двох структур із застосуванням гібридного алгоритму показані на рис. 6 та 7.

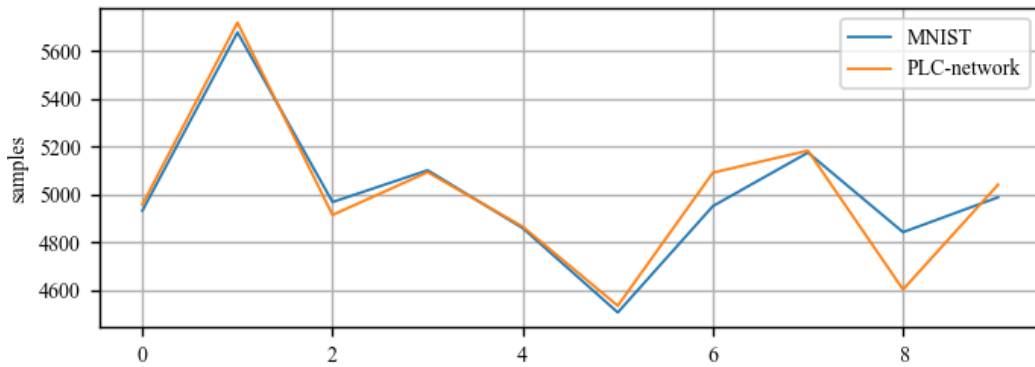


Рис. 6. Результати валідації із застосуванням гібридного алгоритму для STRUCTURE2

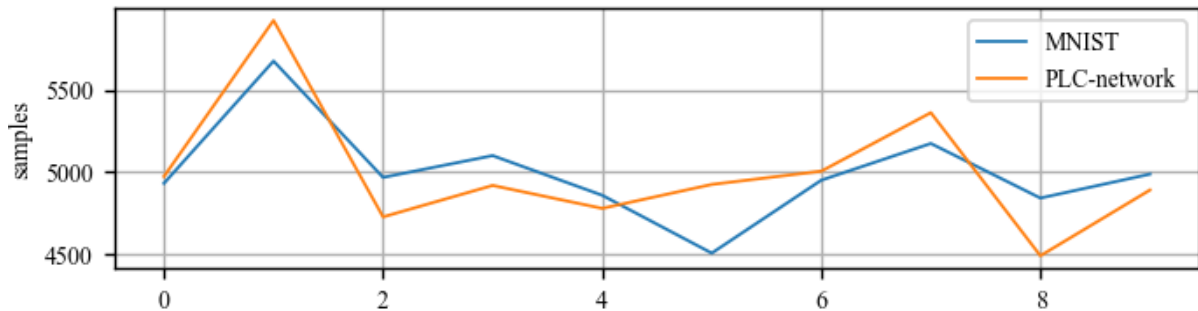


Рис. 7. Результати валідації із застосуванням гібридного алгоритму для STRUCTURE3

У параметричному вигляді результати порівняння результатів використання гібридного алгоритму приведені у табл. 3.

Таблиця 3

Результати порівняльного аналізу результатів кластеризації за допомогою гібридного алгоритму

Гіперпараметри			Параметри рішень			
Structure	n init	max iter	alpha_opt	Dmean %	Dmax %	Час, с
STRUCTURE2	2	10	0.7666	1.208	4.710	453
STRUCTURE3	2	10	1.222	3.808	8.381	423

Аналіз даних у табл. 3 дозволяє зробити наступні висновки:

- зменшення розмірності фактор-простору з 256 координат (STRUCTURE2) до меншої розмірності 128 координат (STRUCTURE3) не призводить до суттєвого пришвидшення алгоритму, втім дуже чутливо впливає на погіршення точності;
- для варіанту STRUCTURE2 отримано найкращий результат за обома критеріями точності: все-ж таки середня відносна похибка приблизно 1% може вважатись дуже прийнятним результатом для вирішення задачі навчання без вчителя.

Обговорення, висновки та рекомендації

Пропонований принцип далекодії-близькодії дозволяє суттєво зменшити кількість параметрів штучних нейронних мереж, які потрібно навчати. Тому в усіх варіантах порівняння алгоритмів ПДБ-мережі дозволяють значно зменшити час вирішення задачі.

Найкращі результати за показниками точності отримано при використанні гібридного алгоритму, який поєднує переваги ПДБ-мереж (швидкість та простота навчання) з можливостями класичного алгоритму k-means.

В цілому, з урахуванням варіабельності зразків рукописних цифр (хто ж їх вчив так писати ці цифри?) всі розглянуті алгоритми можна вважати досить працездатними для конкретної задачі, що вирішувалась. Втім, пропонований гібридний алгоритм показав дуже високі результати як по критеріям точності, так і по критерію швидкості. У найкращому випадку час вирішення задачі склав менше 8 хвилин, тоді як за критерієм середньої відносної кількості помилок отримана оцінка приблизно 1,2%. Тобто, у середньому даний алгоритм дає близько 98,8% вірних рішень.

Можна також відмітити, що всі алгоритми чутливі до первинного вибору гіперпараметрів, що не є якимось новим відкриттям.

Отримані позитивні результати застосування алгоритмів ПДБ-мереж в цілому призводять до висновку щодо доцільності подальшого розвитку теорії та методів нелінійних дискретних згорток, а також розробці високошвидкісного програмного забезпечення, що реалізує принцип далекодії-близькодії.

Література

1. Одегов М.А. Обґрунтування швидких алгоритмів класифікації на множинах BIG DATA за критеріями надійності і продуктивності / М.А. Одегов, М.М. Гаджиев, Л.М. Буката, Л.В. Глазунова, М.В. Кочеткова // Інфокомунікаційні та комп'ютерні технології. - №1, 2023. - С. 148 - 160. DOI: <https://doi.org/10.36994/2788-5518-2023-01-05-16>.
2. М. Одегов, Ю. Бабіч, Д. Багачук, М. Кочеткова, Я. Петрович : Методика критеріїв сум у задачах тестування незалежності послідовностей випадкових чисел // Інфокомунікаційні технології та електронна інженерія: Львів, №2, 2023. С. 20-22. DOI: <https://doi.org/10.23939/ictee2023.02.020>.
3. Одегов М.А. Методика двокомпонентного експрес-тестування незалежності послідовностей псевдовипадкових чисел / М.А. Одегов, Ю.О. Бабіч, Д.Г. Багачук, М.В. Кочеткова, Я.О. Петрович // Міжнародний наук.-техн. журнал "Вимірювальна та обчислювальна техніка в технологічних процесах". - Хмельницькій. - 2023. - № 4. С. 64 - 73. DOI: <https://doi.org/10.31891/2219-9365-2023-76-8>.
4. Одегов М.А. Принцип далекодії-близькодії у задачах структуризації та навчання штучних нейронних мереж / М.А. Одегов, Ю.О. Бабіч // Вісник Хмельницького національного університету. Технічні науки. - 2024. - №3. - С. 357 - 365. DOI 10.31891/2307-5732-2024-337-3-54.
5. J. J. Zhou, "Research on the Complexity Characteristics of Convolutional Neural Networks," 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2023, pp. 402-405, doi: 10.1109/ITOEC57671.2023.10291768.
6. C. -C. Hsu and C. -W. Lin, "Unsupervised convolutional neural networks for large-scale image clustering," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 390-394, doi: 10.1109/ICIP.2017.8296309.
7. . Xiaodong, G. Wendong and Y. Jie, "Handwritten Yi Character Recognition with Density-Based Clustering Algorithm and Convolutional Neural Network," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 2017, pp. 337-341, doi: 10.1109/CSE-EUC.2017.67.
8. W. -T. Su, C. -C. Hsu, Z. Huang, C. -W. Lin and G. Cheung, "Joint Pairwise Learning and Image Clustering Based on a Siamese CNN," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 1992-1996, doi: 10.1109/ICIP.2018.8451224.
9. Sona Taheri, Adil M. Bagirov, Nargiz Sultanova, Burak Ordin, Robust clustering algorithm: The use of soft trimming approach, Pattern Recognition Letters, Volume 185, 2024, Pages 15-22, ISSN 0167-8655, <https://doi.org/10.1016/j.patrec.2024.06.032>.
10. Wenbang Yang, Zaijun Zhang, Yong Zhao, Ye Gu, Lu Huang, Jianchuan Zhao, CABGSI: An efficient clustering algorithm based on structural information of graphs, Journal of Radiation Research and Applied Sciences, 2024, 101040, ISSN 1687-8507, <https://doi.org/10.1016/j.jrras.2024.101040>.
11. Machine Learning in Python scikit-learn [Електронний ресурс] – Режим доступу: <https://scikit-learn.org/stable/>

References

1. Odegov M.A. Obgruntuvannya shvidkih algoritmiv klasifikaciyi na mnozhinah BIG DATA za kriteriyami nadijnosti i produktivnosti / M.A. Odegov, M.M. Gadzhiev, L.M. Bukata, L.V. Glazunova, M.V. Kochetkova // Infokomunikacijni ta komp'yuterni tehnologiyi. - №1, 2023. - S. 148 - 160. DOI: <https://doi.org/10.36994/2788-5518-2023-01-05-16>.
2. M. Odegov, Yu. Babich, D. Bagachuk, M. Kochetkova, Ya. Petrovich . Metodika kriteriyiv sum u zadachah testuvannya nezalezhnosti poslidovnostej vipadkovih chisel // Infokomunikacijni tehnologiyi ta elektronna inzheneriya: Lviv, №2, 2023. S. 20-22. DOI: <https://doi.org/10.23939/ictee2023.02.020>.

3. Odegov M.A. Metodika dvokomponentnogo ekspres-testuvannya nezalezhnosti poslidovnostej psevdovipadkovih chisel / M.A. Odegov, Yu.O. Babich, D.G. Bagachuk, M.V. Kochetkova, Ya.O. Petrovich // Mizhnarodnij nauk.-tehn. zhurnal "Vimiryvalna ta obchislyvalna tehnika v tehnologichnih procesah". - Hmelnickij. - 2023. - № 4. S. 64 - 73. DOI: <https://doi.org/10.31891/2219-9365-2023-76-8.3>. Gerasimchuk O.I. Generatori psevdovipadkovih chisel, yih zastosuvannya, klasifikaciya, osnovni metodi pobudovi i ocinka yakosti / O.I.Gerasimchuk, V.M.Maksimovich // Naukovo - tehnicnij zhurnal "Zahist informaciyi", № 3, 2003. S. 29-36. DOI: <https://doi.org/10.18372/2410-7840.5.4270>.
4. Odehov M.A. Pryntsyp dalekodii-blyzkodii u zadachakh strukturyzatsii ta navchannia shtuchnykh neironnykh merezh / M.A. Odehov, Yu.O. Babich // Visnyk Khmelnytskoho natsionalnoho universytetu. Tekhnichni nuky. - 2024. - №3. - S. 357 - 365. DOI 10.31891/2307-5732-2024-337-3-54.
5. J. J. Zhou, "Research on the Complexity Characteristics of Convolutional Neural Networks," 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2023, pp. 402-405, doi: 10.1109/ITOEC57671.2023.10291768.
6. C. -C. Hsu and C. -W. Lin, "Unsupervised convolutional neural networks for large-scale image clustering," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 390-394, doi: 10.1109/ICIP.2017.8296309.
7. . Xiaodong, G. Wendong and Y. Jie, "Handwritten Yi Character Recognition with Density-Based Clustering Algorithm and Convolutional Neural Network," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 2017, pp. 337-341, doi: 10.1109/CSE-EUC.2017.67.
8. W. -T. Su, C. -C. Hsu, Z. Huang, C. -W. Lin and G. Cheung, "Joint Pairwise Learning and Image Clustering Based on a Siamese CNN," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 1992-1996, doi: 10.1109/ICIP.2018.8451224.
9. Sona Taheri, Adil M. Bagirov, Nargiz Sultanova, Burak Ordin, Robust clustering algorithm: The use of soft trimming approach, Pattern Recognition Letters, Volume 185, 2024, Pages 15-22, ISSN 0167-8655, <https://doi.org/10.1016/j.patrec.2024.06.032>.
10. Wenbang Yang, Zaijun Zhang, Yong Zhao, Ye Gu, Lu Huang, Jianchuan Zhao, CABGSI: An efficient clustering algorithm based on structural information of graphs, Journal of Radiation Research and Applied Sciences, 2024, 101040, ISSN 1687-8507, <https://doi.org/10.1016/j.jrras.2024.101040>.
11. Machine Learning in Python scikit-learn [Електронний ресурс] – Режим доступу: <https://scikit-learn.org/stable/>