https://doi.org/10.31891/2307-5732-2025-359-126
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
ГАДЖИЄВ МАТІН
Державний університет інтелектуальних технологій і зв'язку
http://orcid.org/0000-0001-7280-3863
e-mail: gadjievmm@ukr.net
ПЕРЕКРЕСТОВ ІГОР
Державний університет інтелектуальних технологій і зв'язку
https://orcid.org/0009-0007-3805-8143
e-mail: perekrestov.igor@gmail.com
Державний університет інтелектуальних технологій і зв'язку
ЩЕРБА СЕРГІЙ
https://orcid.org/ 0009-0004-2457-9122
e-mail: sergeyscherba2@gmail.com

АЛГОРИТМИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ ПДБ-МЕРЕЖ

У попередніх роботах було висунуто принцип далекодії-близькодії (ПДБ) для структуризації штучних нейронних мереж. Суть цього принципу полягає у тому, щоб задавати перехідні матриці перетворень між шарами мережі практично як статичні об'єкти з технократичних міркувань: щодалі нейрон наступного шару від нейрону попереднього шару, то менше сигнал з виходу попереднього нейрону впливає на наступний нейрон.

Важливими характеристиками ПДБ-мереж є параметри умовної відстані між послідовними шарами мережі. У роботі пропонується радіальна структура ПДБ-мережі. Доводиться, що якщо кількості нейронів попереднього та наступного шарів є парними числами, то виконуються умови балансу метрик і балансу впливів нейронів попереднього шару на нейрони наступного шару мережі.

Умовні відстані між шарами ставляться в залежність від кількості нейронів попереднього шару N. Таку залежність принципово можна моделювати будь-якою монотонно зростаючою функцією від числа N. У даній роботі прийнята модель ступеневої функції з показником Q, який і є єдиним гіперпараметром ПДБ-мережі. Виконаний порівняльний аналіз за допомогою ПДБ-мережі та метода К-середніх показує перевагу першого як за показником точності, так і за показником швидкості після навчання.

Ключові слова: *штучні нейронні мережі, принцип далекодії-близькодії, алгоритм k-середніх, гіперпараметри, критерії оптимальності.*

NICK ODEGOV, HADZHYIEV MATIN, PEREKRESTOV IHOR, SERHII SHCHERBA
State University of Intellectual Technologies and Telecommunications

ALGORITHMS FOR OPTIMIZATION OF LCR-NETWORK HYPERPARAMETERS

This paper continues the development of methods for solving problems involving ultra-large data sets (Big Data class problems). The problems are solved with respect to two inherently conflicting criteria: accuracy and speed.

In previous studies, the long-and-close-range (LCR) principle was introduced for structuring artificial neural networks. According to this principle, the transition matrices between network layers are treated as nearly static objects based on technocratic reasoning: the farther a neuron in the next layer is from a neuron in the previous layer, the less influence it receives from the output of that previous neuron. Rather than optimizing hundreds of thousands or even millions of transition matrix elements, as in conventional feedforward networks, the proposed LCR networks require the optimization of only a few hyperparameters and activation function settings.

A key characteristic of LCR networks is the parameterization of the conditional distance between consecutive layers. This work introduces a radial structure for LCR networks and proves that when both the previous and subsequent layers contain an even number of neurons, the metric balance and influence balance conditions are satisfied.

The conditional distances are modeled as a function of the number of neurons in the previous layer, N, and can generally follow any monotonically increasing function. In this study, a power-law function is adopted, with the exponent Q serving as the sole hyperparameter of the LCR network in this configuration. Thus, the model optimizes only one scalar parameter instead of tuning numerous matrix coefficients.

A comparative analysis involving LCR networks and the classical k-means algorithm demonstrates that the proposed network achieves better performance both in terms of classification accuracy and processing speed after training.

Keywords: *artificial neural networks, long-and-close-range principle, k-means algorithm, hyperparameters, optimality criteria.*

Стаття надійшла до редакції / Received 01.09.2025
Прийнята до друку / Accepted 15.11.2025

Аналіз джерел та постановка проблеми

У наукових працях кафедри ІПЗ ДУІТЗ значна увага приділяється розробці ефективних алгоритмів для обробки надвеликих масивів даних [1–3], що належать до класу задач Big Data. Чітке розмежування між «звичайними» даними та Big Data залежить від характеру конкретного практичного завдання. Втім, у конкретній задачі ефекти Big Data проявляються тоді, коли надійні та продуктивні алгоритми за показником точності (accuracy) не можуть вирішити задачу за практично прийнятний час. У таких випадках треба приймати деякий компроміс між точністю та швидкістю алгоритмів.

Алгоритми, що розглядаються у роботах [4, 5] були перевірені на відомому наборі рукописних цифр MNIST. Приклади рукописних цифр з цього набору наведені подано на рис. 1.

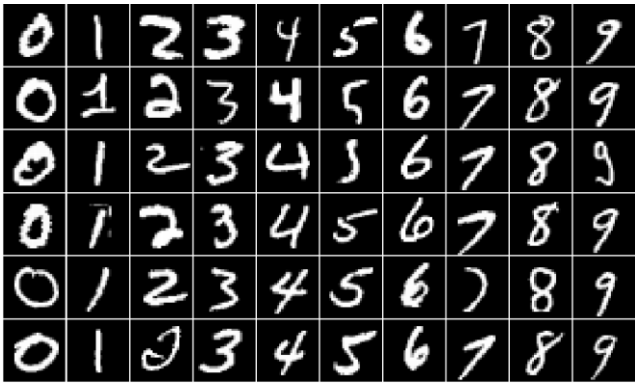

Рис. 1. Приклади зображень рукописних цифр набору MNIST

Як можна побачити з рис. 1, накреслення цифр дуже відрізняються. Деякі зразки суттєво відрізняються навіть від типових представників свого класу. Так, сімка у першому рядку скоріше нагадує одиницю, двійка у другому рядку скоріше нагадує шістку, що дзеркально відображена. Двійка у четвертому рядку скоріше нагаду сімку, а в останньому рядку – дуже схожа на нуль. П'ятірка у третьому рядку скоріше нагадую трійку, або одиницю. Окрім того, цифри мають різний кут нахилу, а також різну інтенсивність пікселів (хтось, мабуть, писав тонкою кульковою ручкою, а хтось фломастером). Таким чином, задача розпізнавання цих образів є дуже складною для будь-якого алгоритму.

Відмітимо, що з використанням набору MNIST навіть організовані міжнародні змагання алгоритмів [6]. Автори досліджень при цьому дають найкращі результати навіть більше 99% правильних рішень на тестовому наборі даних Враховуючи велику варіативність написання цифр (рис. 1), настільки високі результати видаються майже нереалістичними. Втім, у даному конкурсі жодним чином не враховується час навчання алгоритмів, а також час розпізнавання тестового набору. Тестування алгоритмів на наборі MNIST, як на полігоні, є темами багатьох наукових робіт. Але, у значній більшості ці роботи спрямовані на визначенні лише показника точності (відсотка вірних рішень). Проте, фактор часу при цьому залишається ніби мало значущим. Серед невеликої кількості робіт, де все ж таки враховується параметр часу, можна назвати статтю [7]. У цій роботі параметр точності досягає 98%, час навчання алгоритму та розпізнавання мають порядок десятків секунд. Втім, навчання гіперпараметрів складає вже дні та тижні.

У роботах [4, 5] штучні нейронні мережі, побудовані за принципом далекодії-близькодії (надалі ПДБ-мережі) застосовувались для вирішення задачі кластеризації набору MNIST. Отримані результати показали працездатність цих алгоритмів, а також високу ефективність за показником швидкості.

Метою даною роботи є обґрунтування критеріїв та алгоритмів оптимізації гіперпараметрів ПДБ-мереж для вирішення задачі класифікації.

Визначення гіперпараметрів ПДБ-мережі

Штучні нейронні мережі можуть розглядатись як дуже примітивна та зовсім некоректна модель нейронів мозку та зв'язків між нейронами. Парадигма ПДБ-мереж полягає у застосуванні «технократичної» моделі передачі даних між різними шарами мережі. Надалі, все ж таки, будемо використовувати термінологію, що є на даний час сталою: нейрон – елемент шару перетворювань, мережа – штучна нейронна мережа (ШНМ) і т. п.

Основна ідея структуризації ПДБ-мереж полягає у тому, щоб практично відмовитись від навчання перехідних матриць між сусідніми шарами ШНМ. Наприклад, за наявності лише 1000 нейронів у кожному з двох сусідніх шарів необхідно навчати матрицю з мільйоном елементів. Якщо структура та параметри перехідною матриці вже визначені, то можна оптимізувати лише 1000 параметрів функцій активації попереднього шару.

Перехідні матриці між шарами задаються практично як як майже статичні шаблони, які мають дуже невелику кількість параметрів (у даному класі моделей - гіперпараметрів), що дуже швидко навчаються. Для визначення структури таких матриць у роботах [4, 5] запропоновано принцип далекодії-близькодії, який можна сформулювати у таких тезах:

– щодалі далі нейрон попереднього шару ШНМ від нейрону наступного шару, тим менший вплив перший чинить на другий;

– сума відстаней від кожного нейрону попереднього шару ШНМ до всіх нейронів наступного шару має бути однаковою для всіх нейронів попереднього шару (умова балансу метрик прямого напрямку);

– сума відстаней від кожного нейрону наступного шару ШНМ до всіх нейронів попереднього шару має бути однаковою для всіх нейронів наступного шару (умова балансу метрик зворотного напрямку);

– сума впливів (коефіцієнтів передачі) нейронів попереднього шару ШНМ на нейрони наступного шару має бути однаковою для всіх нейронів попереднього шару (умова енергетичного балансу мережі).

При виконанні цих умов ШНМ знаходиться у рівноважному стані якщо на вхід нейронів попереднього шару взагалі не подаються сигнали, або на всі нейрони подаються однакові сигнали. Мережа виходить із рівноважного стану лише за умови, якщо сигнали на входах нейронів попереднього шару різні.

Одна з можливих структур ПДБ-мережі, яка задовольняє даним умовам, показана на рис. 2. На даному рисунку показано, що нейрони сусідніх шарів розміщені з рівними кроками для кожного шару:

$$\propto_n = \frac{2\pi n}{N}\ ,\ n = \overline{0, N-1}\ \ ;\ \ \beta_m = \frac{2\pi m}{M}\ ,\ m = \overline{0, M-1}\ . \quad (1)$$

Відстань між нейронами шарів у полярній системі координат можна визначити через відносну відстань $q = (R - r)/r$ між шарами [4]:

$$\Delta R(n, m) = 2r^2(1 + q)[1 - \cos(\alpha_n - \beta_m)] + q^2 r^2\ , \quad (2)$$

звідки мінімальна та максимальна відстані визначаються так:

$$\Delta R_{min} = q^2\ r^2\ ;\ \ \Delta R_{max} = 4r^2(1 + q) + q^2\ r^2\ . \quad (3)$$

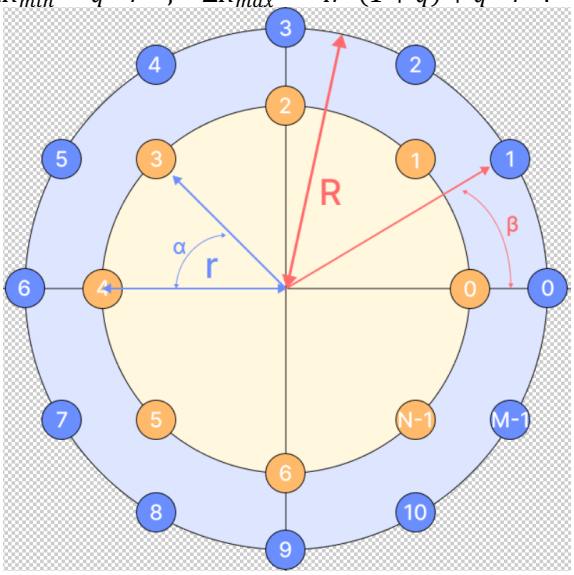

Рис. 2. Варіант радіальної структури ПДБ-мережі для *N*=8, *M*=12

Покажемо, що за певних умов у структурі за рис. 2 виконуються умови балансу метрик та енергетичного балансу мережі.

Лема про баланс метрик: якщо N та M парні числа, то умова балансу метрик виконується. Суми відстаней між нейронами згідно формули (2) відрізняються лише значенням складової $\cos(\alpha_n - \beta_m)$. Зафіксуємо будь який нейрон попереднього шару з кутом $\alpha = \alpha_n$. Тоді сума тригонометричних функцій у формулі (2) буде визначатись:

$$s(\alpha) = \cos(\alpha) \sum_{m=0}^{M-1} \cos(\beta_m) + \sin(\alpha) \sum_{m=0}^{M-1} \sin(\beta_m)\ . \quad (4)$$

Внаслідок парності числа M для кожного номеру нейрона m завжди можна знайти відповідний нейрон з номером $m2 = m \pm M/2$. Тоді нейрони з номерами m та $m2$ згідно визначень (1) будуть знаходитись у протифазах, тобто $\beta_{m2} = \beta_m \pm \pi$. Звідки витікає, що суми синусів та косинусів у виразі (4) будуть включати пари, суми яких завжди дорівнюють нулю, тобто:

$$\forall m:\ \cos(\beta_m) + \cos(\beta_m + \pi) = 0\ ;\ \ \sin(\beta_m) + \sin(\beta_m) = 0\ ,$$

а сума відстаней від будь якого нейрона попереднього шару до нейронів наступного шару буде складати

$$S(n, M) = Mr^2[2(1 + q) + q^2 r^2].$$

У зворотному напрямку Лема доводиться аналогічно. Наслідком Леми є наступна

Теорема про енергетичний баланс мережі: суми будь-яких монотонних функцій $W[\Delta R(n, m)]$ відстаней від нейронів попереднього шару до нейронів наступного шару є однаковою для всіх нейронів попереднього шару. Справедливе і зворотне твердження. Доказ Теореми безпосередньо витікає із способу доказу Леми. Дійсно, складові сум метрик відрізняються лише порядком нумерації нейронів та кутів.

Тоді і значення функцій $W[\Delta R(n, m)]$ будуть відрізнятись лише порядком нумерації, а їхня сума від перестановки номерів, звісно, не залежить.

Функції $W[\Delta R(n, m)]$ можна умовно назвати функціями впливу. Відомо, що енергетичні та гравітаційні взаємодії зменшуються у зворотній пропорції до квадрату відстаней. Втім, у програмних розробках на даний час прийнята лінеаризована модель для малих відстаней: залежність зворотно до першого ступеня відстані. Тобто, якщо розрахована матриця відстаней (2), то перехідна матриця (матриця впливів) буде визначатись:

$$W[\Delta R(n, m)] = 1/\Delta R(n, m). \quad (5)$$

Для остаточного вирішення задачі визначення гіперпараметрів залишається лише у формулі (2) з якихось додаткових міркувань запропонувати спосіб знаходження параметру q (відносної відстані між шарами).

Якщо умовно прийняти мінімальну відстань між шарами у формулах (3) $\Delta R_{min} = 1$, а максимальну відстань покласти як якусь монотонно зростаючу функцію $F(N)$ від числа нейронів попереднього шару, то параметр q визначатиметься як єдиний позитивний корінь квадратного рівняння:

$$[1 - f(N)]q^2 + 4q + 4 = 0 \quad \Rightarrow \quad q = \frac{2 + 2\sqrt{f(N)}}{f(N) - 1}. \tag{6}$$

Якщо кількість нейронів N дуже велика, то приблизне рішення рівняння (6): $q \approx 2/\sqrt{f(N)}$. Наприклад, якщо $f(N) = N^2$, то $q \approx 2/N$. Далі під параметричним сімейством функцій $f(N)$ розуміються ступеневі функції: $f(N) = N^Q$. Показник ступеню Q і є єдиним числовим гіперпараметром мережі. Тут лише зауважимо, що «навчити» значення лише одного числа Q можна швидше, ніж тисячі або мільйони параметрів перехідних матриць у ШНМ прямого розповсюдження.

Алгоритм оптимізації гіперпараметру *Q*

Оптимізацію алгоритмів вирішення задачі класифікації можна декомпозувати на дві незалежні (або частково залежні) задачі:

– визначення оптимальних параметрів алгоритму так, щоб після перетворень відстані образів вхідних прототипів одного й того ж класу були якомога більш однаковими;

– визначення цих чи інших параметрів алгоритму так, щоб вихідні образи з різних класів були якомога більш віддаленими.

Наведемо лише один з варіантів оптимізації гіперпараметру Q лише для вирішення першої з цих задач. У даному випадку логічним є застосування показників, які так чи інакше враховують ентропію розподілень кількості зразків в одному класі. Одним з таких показників може бути коефіцієнт модальності, який визначається наступним чином.

Для кожного з K класів визначається однакова кількість кластерів (значень різних центроїдів) C. Наприклад, для кожного класу визначається C = 1000 кластерів. Спочатку ці кластери заповнюються різними випадковими значеннями зразків, однакових класів для кожного кластеру. Далі визначаються розмічені зразки, що є найближчими для одного з кластерів конкретного класу та підраховується кількість зразків одного класу, що потрапили у той чи інший кластер. Далі масив цих кількостей $N_{k,c}$ для кожного класу K ранжується таким чином, щоб $N_{k,c} \geq N_{k,c+1}$, тобто у порядку спадання.

Визначається параметр концентрації z – відсоток або частка кластерів з найбільшою кількістю зразків. Наприклад, якщо кількість кластерів C = 1000, а заданий параметр концентрації $z = 0.2$, то кількість зразків найбільшої концентрації буде розраховуватись для 20 кластерів. Для кожного класу k визначається класовий коефіцієнт концентрації як відношення кількості зразків $N_{z,c}$ у кластерах з номерами $Nmode(z,k) = [z \cdot C]$ до загальної кількості зразків у класі $Nsumm(k)$: $Conc(k) = Nmode(z,k)/Nsumm(k)$. Узагальнюючий показник – коефіцієнт модальності визначається як середнє значення класових коефіцієнтів концентрації для всіх класів:

$$ModCoeff = \sum_k Conc(k)/K. \tag{7}$$

По суті, коефіцієнт модальності характеризує відношення кількості зразків в заданій області поблизу моди розподілення цієї кількості у кластерах. З визначення (7) безпосередньо витікає критерій оптимізації гіперпараметру Q ПДБ-мережі:

$$Qopt = \underset{Q}{\operatorname{argmax}}\, ModCoeff(Q). \tag{8}$$

Значення коефіцієнту модальності (7) та визначення оптимального значення (8) гіперпараметра Q, зрозуміло, буде залежати від кількості нейронів у попередньому та наступному шарі ПДБ-мережі. На першій ступені перетворень кількість нейронів попереднього шару (вхідного шару нейронів) завжди дорівнює кількості пікселів у зображеннях цифр ($N = 28 \times 28 = 784$), а кількість нейронів M наступного (у даному випадку - вихідного) шару може змінюватись в залежності від плану чисельного експерименту – плану тестування алгоритмів.

План порівняльного тестування алгоритмів

У даному випадку кількість класів дорівнює кількості цифр (K = 10). Кількість кластерів C для кожного класу у числових тестах взята або 100, або 200. Тестування проводилось лише за допомогою CPU (прискорювачі за допомогою GPU не використовувались). В якості єдиного програмного прискорювача для повільних програм на мові Python використано лише скомпільовану на низькому рівні бібліотеку NumPy.

Порівняльний аналіз виконано для двох алгоритмів, заснованих на методі К-середніх (K-means), який зазвичай використовується для вирішення задач кластеризації. Втім, у задачах класу Big Data він дає значну перевагу за параметром швидкості [1].

Навчання першого (базового) алгоритму, якій надалі називається просто K-means, складається з наступних кроків:

– із набору тренувальних розмічених зразків відокремлюється якась відносно невелика сукупність з кількістю зразків Tr, наприклад, Tr =10000 або Tr =20000 (навчальна вибірка);

– задається кількість кластерів C;

– первинно кластери заповнюються випадковими зразками з відповідними до класу мітками;

– ці зразки надалі використовуються як первинні центроїди кластерів;

– у циклі по кількості зразків у навчальній вибірці виконується операція уточнення центроїдів за рекурентною формулою, а саме:

якщо зразок S є найближчим до певного кластеру і при цьому мітка класу дорівнює мітці зразка S та до цього положення центроїду Center(L) було уточнено з попереднім врахуванням цих зразків L разів, то нове положення центроїду кластера буде визначатись за формулою:

$$Center(L+1) = [L \cdot Center(L+1) + S]/L\,.$$

Після навчання (уточнення положень центроїдів) вирішується суто задача класифікації на відокремленому наборі зразків. У даному дослідженні цей тестовій набір включає 10000 зразків. Він також є розміченим і тому можна визначити відсоток вірної та помилкової класифікації. Перший з цих показників і буде параметром точності (accuracy).

Алгоритм із застосуванням ПДБ-мереж відрізняється від попереднього лише проміжним циклом навчання за критерієм (7) і тим, що класифікуються не самі зразки, а їхні образи за допомогою перехідної матриці W (5). У конкретних числових експериментах оптимізація параметру Q виконувалась найпростішим та найточнішим методом повного перебору можливих значень, наприклад (у термінах списків мови Python):
Q = [1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0]. Один з типових графіків залежності коефіцієнта модальності від параметра відстані між шарами мережі Q наведено на рис. 3.

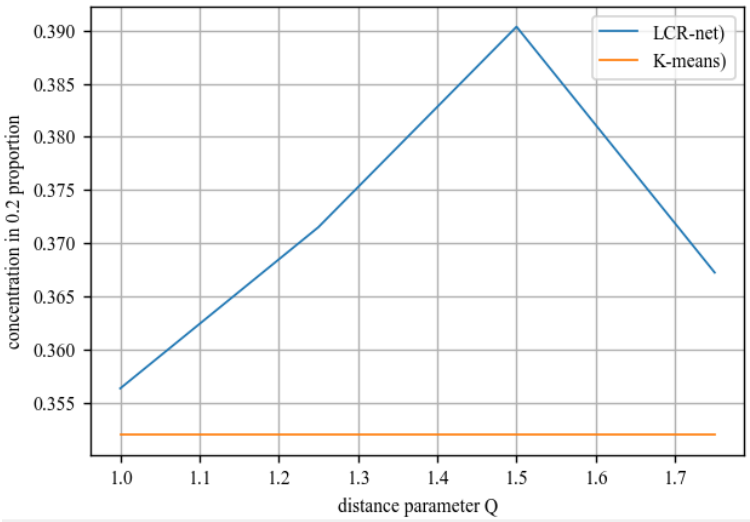

Рис. 3. Залежність коефіцієнту модальності від параметра відстані між шарами мережі Q

У табл. 1 наведено деякі приклади результати порівняльного тестування визначених алгоритмів. У цей таблиці використано позначку *dMC* – відсоток підвищення значення коефіцієнту *ModCoeff* у навченій мережі порівняно із цим коефіцієнтом для вхідних зразків.

Таблиця 1

Результати порівняльного аналізу результатів класифікації

Tr	*C*	*dMC*	K-means		LCR-net			
			Accuracy	Time, s	*M*	*Qopt*	Accuracy	Time, s
5000	100	8.13	90.44	60	128	1.5	91.02	44
5000	100	8.01	90.44	60	256	1.5	91.08	46
5000	100	8.63	90.44	60	512	1.5	91.18	50
10000	100	6.84	90.64	60	256	1.75	91.53	51
10000	100	7.22	90.64	60	512	1.5	92.09	57
10000	200	9.78	92.28	134	256	1.25	93.36	98
10000	200	9.78	92.28	134	512	1.5	94.18	105

Обговорення, висновки та рекомендації

Аналіз даних у табл. 1 показує, що у наведених прикладах тестування алгоритм LCR-net переважає класичний алгоритм K-means як за критерієм точності, так і за критерієм швидкості класифікації. Останній ефект пояснюється тим, що у використаних варіантах структури ПДБ-мереж здійснюється редукція первинного простору розмірності 784 фактори у простір меншої розмірності. Більш цікавим є збільшення точності рішень. Даний ефект пояснюється тим, що алгоритм навчання ПДБ-мережі збільшує концентрацію зразків біля моди розподілення. Ефективність такого навчання зводиться до того, що коефіцієнт модальності збільшився майже на 10%.

Аналіз графіку на рис. 3 показує (а також інших аналогічних графіків), показують, що залежність коефіцієнту модальності від параметру дистанції між шарами є унімодальною функцією. Дана обставина дозволяє ефективно використовувати будь які послідовні методи оптимізації, наприклад градієнтний чи метод половинного ділення або золотого перетину.

Перевага пропонованого алгоритму над класичним алгоритмом K-means у даному випадку є несуттєвою. Втім, треба врахувати, що у даній роботі повноцінного навчання ПДБ-мережі поки що не відбувалось: навчався лише єдиний гіперпараметр. Навчання параметрів нейронів шарів перетворень є предметом наступних досліджень.

Література

1. Одегов М.А. Обґрунтування швидких алгоритмів класифікації на множинах BIG DATA за критеріями надійності і продуктивності / М.А. Одегов, М.М. Гаджиєв, Л.М. Буката, Л.В. Глазунова, М.В. Кочеткова // Інфокомунікаційні та комп'ютерні технології: Київ. - №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. Одегов М.А. Порівняльний аналіз алгоритмів кластеризації / М.А. Одегов, Д.Г. Багачук, І.С. Перекрестов, Я.О. Петрович // Вісник Хмельницького національного університету. Технічні нуки. - 2024. - №4. - С. 406-413. DOI 10.31891/2307-5732-2024-339-4-61.

6. Kaggle.com: Digit Recognizer / [Електронний ресурс] – Режим доступу: https://www.kaggle.com/competitions/digit-recognizer.

7. Ekaterina Plesovskaya, Sergey Ivanov, Hierarchical Classification on the MNIST Dataset Using Truncated SVD and Kernel Density Estimation, Procedia Computer Science, Volume 212, 2022, Pages 368-377, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.11.021.

References

1. Odegov M.A. Obgruntuvannya shvidkih algoritmiv klasifikaciyi na mnozhinah BIG DATA za kriteriyami nadijnosti i produktivnosti / M.A. Odegov, M.M. Gadzhiyev, 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 "Vimiryuvalna ta obchislyuvalna tehnika v tehnologichnih procesah". - Hmelnickij. - 2023. - № 4. S. 64 - 73. DOI: https://doi.org/10.31891/2219-9365-2023-76-8.

4. Odegov M.A. Princip dalekodiyi-blizkodiyi u zadachah strukturizaciyi ta navchannya shtuchnih nejronnih merezh / M.A. Odegov, Yu.O. Babich // Visnik Hmelnickogo nacionalnogo universitetu. Tehnichni nauki. - 2024. - №3. - S. 357 - 365. DOI 10.31891/2307-5732-2024-337-3-54.

5. Odegov M.A. Porivnyalnij analiz algoritmiv klasterizaciyi / M.A. Odegov, D.G. Bagachuk, I.S. Perekrestov, Ya.O. Petrovich // Visnik Hmelnickogo nacionalnogo universitetu. Tehnichni nuki. - 2024. - №4. - S. 406-413. DOI 10.31891/2307-5732-2024-339-4-61.

6. Kaggle.com: Digit Recognizer / [Електронний ресурс] – Режим доступу: https://www.kaggle.com/competitions/digit-recognizer.

7. Ekaterina Plesovskaya, Sergey Ivanov, Hierarchical Classification on the MNIST Dataset Using Truncated SVD and Kernel Density Estimation, Procedia Computer Science, Volume 212, 2022, Pages 368-377, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.11.021.