

ОДЕГОВ МИКОЛА

Державний університет інтелектуальних технологій і зв'язку

<https://orcid.org/0000-0001-5526-2487>e-mail: onick_64@ukr.net

БАБІЧ ЮРІЙ

Державний університет інтелектуальних технологій і зв'язку

<https://orcid.org/0000-0002-7888-7591>e-mail: babich159@gmail.com

ПРИНЦИП ДАЛЕКОДІЇ-БЛИЗЬКОДІЇ У ЗАДАЧАХ СТРУКТУРИЗАЦІЇ ТА НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Класичні штучні нейронні мережі у загальному випадку потребують навчання значної кількості параметрів перехідних матриць між сусідніми шарами. Основна ідея роботи полягає в тому, щоб ці матриці задавати один раз та жорстко із якихось «розумних міркувань». Тоді навчати буде потрібно лише нейрони мережі. В якості такого «розумного міркування» у даній роботі використовується принцип далекодії-близькодії. Суть цього принципу полягає у тому, що нейрони попереднього шару впливають на нейрони наступного шару мережі тим більше, чим вони ближчі за топологією мережі. В даній роботі пропонується використовувати радіальну структуру топології. Такий спосіб визначення геометричного розташування шарів забезпечує умови балансу мережі, а саме: сукупність впливів нейронів попереднього шару на нейрони сусіднього шару є константою незалежно від умовних номерів нейронів.

За математичною сутністю пропонується штучні нейронні мережі, засновані на принципі далекодії-близькодії можна віднести до специфічного підкласу нелінійних згорткових мереж. Нелінійні згортки реалізуються за допомогою ядерних дискретних перетворень, де ядрами перетворень є перехідні матриці зв'язків між сусідніми шарами.

В якості функцій активації розглянуто параметричні сигмоїди, які мають лише один вільний параметр – коефіцієнт нелінійності.

Розроблені алгоритми та програми застосовано для вирішення задачі навчання без вчителя – задачі кластеризації. В якості тестового набору даних обрано відомий набір рукописних цифр MNIST. Задачу вирішено на звичайному комп'ютері з використанням лише CPU (GPU не використовувалось).

Результати валідації отриманого розподілу 50 000 зразків набору MNIST по 1000 кластерам виявились дуже обнадійливими. А саме: час вирішення задач навчання та суто кластеризації складає менше 10 хвилин, а точність вірного віднесення до кластерів на етапі валідації досягає 97%.

Ключові слова: штучні нейронні мережі, принцип далекодії-близькодії, навчання без вчителя, нелінійні згорткові мережі, параметричний сигмоїд, перехідні матриці.

ODEGOV NICK, BABICH YURI

State University of Intellectual Technologies and Telecommunications

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

Classical artificial neural networks in the general case require learning a significant number of transition matrix parameters between adjacent layers. The main idea of the work is to set the matrix once and rigidly according to some "reasonable thoughts". Then only the neurons of the network will need to be trained. This work uses the principle of long-and-close-range action in the kind of the "reasonable thoughts". The essence of this principle is that the closer layers are to each other stronger is the influence between neurons. In the current work, it is proposed to use the radial structure of the topology. This way of determining the geometric arrangement of the layers ensures the conditions of the balance of the network, namely: the set of effects of the neurons of the previous layer on the neurons of the neighboring layer is a constant regardless of the conditional numbers of neurons.

According to the mathematical essence, the proposed artificial neural networks based on the long-and-close-range action principle can be classified as a specific subclass of nonlinear convolutional networks. Nonlinear convolutions are implemented with the help of core discrete transforms, where the transition matrices of connections between adjacent layers are the cores of transformations.

As activation functions, parametric sigmoids are considered, which have only one free parameter i.e. the nonlinearity coefficient.

The developed algorithms and programs are applied to solve the problem of unsupervised learning, namely the problem of clustering. The well-known set of handwritten digits MNIST was chosen as the test data set. The problem was solved on a regular computer using only the CPU (GPU was not used).

The results of the validation of the obtained distribution over 50,000 samples of the MNIST set on 1000 clusters yielded very encouraging results. The time for solving the learning and pure clustering tasks is less than 10 minutes, and the accuracy of correct assignment to clusters at the validation stage reaches 97%.

Keywords: artificial neural networks, long-and-close-range action principle, unsupervised learning, nonlinear convolutional networks, parametric sigmoid, transition matrices.

Аналіз джерел та постановка проблеми

Однією із основних тем наукової роботи кафедри ПЗ ДУІТЗ є розробка швидких алгоритмів для вирішення задач класу Big Data [1-3]. Для такого класу задач іноді приходиться дещо нехтувати точністю на користь швидкості. У даній роботі цей напрямок дослідження поширюється на одну із задач штучного інтелекту – задачу кластеризації, тобто на одну із задач навчання без вчителя.

Сучасні тенденції вирішення задачі кластеризації полягають, в основному, у використанні згорткових нейронних мереж [4, 5]. При цьому значна кількість публікацій присвячена кластеризації зображень

(наприклад, [6-8]). У цитованих роботах спільним є використання методу k -середніх, або k -центроїдів 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.

Одному з варіантів вирішення останньої проблеми і присвячується дана робота. Основну ідею вирішення проблеми можна пояснити за допомогою простої схеми двох внутрішніх шарів (рис. 1).

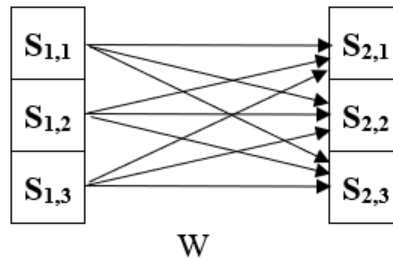


Рис. 1. Внутрішні шари нейронної мережі

На рис.1 визначено два шари: попередній з кількістю нейронів $N = 3$ та наступний з такою ж кількістю нейронів $M = 3$. У даному випадку класичні підходи, засновані на принципах перцептронів Розенблата потребують оптимізації 9-ти елементів матриці вагових коефіцієнтів W , тобто перехідної матриці. І це не дуже складна задача. Однак, якщо у реальній задачі кількість нейронів складатиме $N = M = 1000$ (не дуже велика розмірність для реальних задач), то кількість елементів перехідної матриці W складатиме вже 1 000 000. Тут не стільки філософськи, скільки з технічної точки зору «кількість переходить в якість» зі всіма наслідками: проблемами градієнтів, порядками алгоритмів і т. п.

Наша ідея полягає у тому, щоб замість навчання перехідних матриць W навчати функції активації, та взагалі оператори перетворень нейронів. У наведеному прикладі навчати буде потрібно лише 1000 нейронів попереднього шару замість 1 000 000 елементів матриці W . Тобто, за термінологією, що склалася, «параметри» та «гіперпараметри» як би змінюються місцями за своїм функціоналом.

Деяко забігаючи вперед, відмітимо, що в кінцевому варіанті задачу кластеризації було вирішено на структурі мережі, яку зручно представити у вигляді структури Python:

$$STRUCTURE = ((784, 512), (512, 256), (256, 128), (128, 64)). \tag{1}$$

Зауважимо, що у даній роботі іноді замість багатопверхових формул будуть використовуватись образи мови Python та бібліотеки Numpy. Втім, у класичному варіанті треба було б навчати тим чи іншим способом $784 * 512 * 256 * 128 * 64 = 841\ 813\ 590\ 016$ елементів перехідних матриць. Якщо замість цього навчати нейрони, то їх кількість значно менша: $784 + 512 + 256 + 128 + 64 = 1744$. Виграш у порядку алгоритму за кількістю порівняльних операцій в такому випадку складатиме біля 482 691 278, тобто приблизно у 500 мільйонів разів!

Виграш у швидкості навчання може бути, як бачимо, вирішальним. Але постає проблема: оскільки перехідні матриці треба задавати як стаціонарні об'єкти, то яким чином визначити значення їх елементів? Тут потрібно використовувати якісь, так би мовити, «розумні міркування». У якості таких міркувань пропонується «принцип дальності-близькодії» (ПДБ), обґрунтуванню якого і присвячується дана робота.

Принцип дальності-близькодії

Штучні нейронні мережі часто-густо розглядаються як аналоги, точніше кажучи, алегорії мозку живих істот. Зрозуміло, що це лише примітивні моделі. Втім мозок можна розглядати як електромагнітну систему: інакше, як би лікарі та дослідники отримували енцефалограми? Відомо, що у середньому кожний нейрон мозку людини має безпосередній зв'язок приблизно з 700 іншими нейронами. Справедливо задати питання: чи залежить рівень сигналу, який доходить до нейрона-приймача від нейрона-джерела від фізичної відстані між ними? На жаль, наукові результати у цьому напрямку авторам невідомі. Тому надалі наші «розумні міркування» будуть засновані на технократичній алегорії. Звісно, що у будь-якій технічній системі передачі рівень сигналу на стороні прийому суттєво залежить від відстані між передавачем та приймачем. У більшості випадків розглядаються моделі, коли цей рівень є зворотнім від відстані або від квадрату відстані.

Коротко кажучи, у ПДБ-мережах використовується просте правило: чим далі нейрони наступного шару від нейронів попереднього шару, тим менший вплив перші здійснюють на останні. Але важливо при цьому дотримуватись ще одного принципу: рівноправності нейронних зв'язків. У даній роботі цей принцип розуміється як умова рівної суми відстаней від нейронів попереднього шару до нейронів наступного шару:

$$\sum_{n=0}^{N-1} \Delta R(n, m) = const, \quad \forall m = \overline{0, M-1}, \tag{2}$$

де $\Delta R(n, m)$ – відстань між нейронами шарів; N – кількість нейронів попереднього шару; M – кількість нейронів наступного шару.

Не важко довести, що будь-яка монотонна функція $W[\Delta R(n, m)]$ також буде відповідати умові балансу.

Один з можливих варіантів для дотримання умови (2) має геометричну інтерпретацію, яка умовно показана на рис 2.

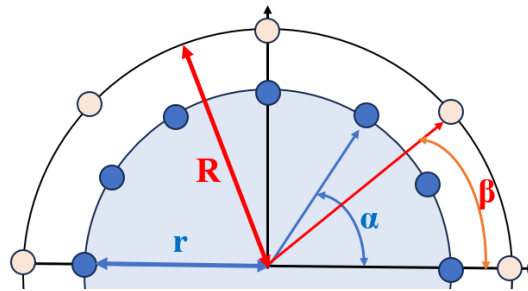


Рис. 2. Геометрична інтерпретація шарів нейронної мережі

На рис. 2 показано, що нейрони шарів розміщуються на відстанях r та R від умовного центру. У полярних координатах з відповідними радіусами та з рівномірно визначеними кутами відповідно до номерів нейронів:

$$\alpha_n = \frac{2\pi n}{N}, \quad n = \overline{0, N-1}; \quad \beta_m = \frac{2\pi m}{M}, \quad m = \overline{0, M-1}. \quad (3)$$

Позначимо відстань між сусідніми шарами: $\Delta r = R - r$. Тоді квадратична відстань між нейронами сусідніх шарів з номерами n та m буде визначатись:

$$\Delta R(n, m) = 2(r^2 + r\Delta r)[1 - \cos(\alpha_n - \beta_m)] + \Delta r^2, \quad (4)$$

або, якщо ввести додатковий параметр q по правилу: $q = \Delta r/r$, то вираз для відстаней (4) можна записати у такому вигляді:

$$\Delta R(n, m) = 2r^2(1 + q)[1 - \cos(\alpha_n - \beta_m)] + q^2 r^2. \quad (5)$$

Якщо умовно покласти $r = 1$, то параметр q буде виконувати роль єдиного гіперпараметру мережі для двох сусідніх шарів. Для визначення цього параметру дослідник може задати функцію $f(N)$ або $f(M)$ в залежності від того, до якого шару треба унормувати відстані – до попереднього, або до наступного. Виконані дослідження показують, що дещо кращі результати дає варіант, коли ця функція визначається для шару з найбільшою кількістю нейронів. Для визначеності будемо вважати, що $N \geq M$. Тоді суть функції $f(N)$ така: якщо найменшу відстань між нейронами сусідніх шарів прийняти за $\Delta R_{min} = 1$, то найбільша відстань буде: $\Delta R_{max} = f(N)$. Наприклад, якщо $f(N) = N$, то максимальна і мінімальна відстані будуть різнитись у N разів. Для конкретної задачі, що вирішувалась, найкращим варіантом була функція $f(N) = N^2$. Відповідно, мінімальне значення елементів перехідних матриць складало $W_{min} = 1/N^2$, а максимальне $W_{max} = 1$.

Із визначення відстаней (5) витікає, що мінімальна та максимальна відстані між нейронами сусідніх шарів визначаються наступним чином:

$$\Delta R_{min} = q^2 r^2; \quad \Delta R_{max} = 4r^2(1 + q) + q^2 r^2. \quad (6)$$

Тоді, з врахуванням визначень функції $f(N)$ та мінімаксних параметрів відстаней (6), зводимо визначення гіперпараметру q до шкільного квадратичного рівняння:

$$[1 - f(N)]q^2 + 4q + 4 = 0, \quad (7)$$

яке має єдиний позитивний корінь відносно вирішення для гіперпараметру q :

$$q = \frac{2 + 2\sqrt{f(N)}}{f(N) - 1}, \quad (8)$$

а вже з урахуванням залежностей виду (8) розраховуються матриці відстаней ΔR за формулою (5). В кінцевому підсумку, визначаються перехідні матриці, що простіше всього виразити у поняттях NumPy Python:

$$W = \frac{1}{\Delta R}. \quad (9)$$

Визначення параметричних функцій активації, умов балансу мережі та нормалізації

У конкретному випадку, що розглядається к даній роботі, ПДБ-мережі застосовуються для кластеризації зразків зображень. Більш конкретно – рукописних цифр з тестового набору MNIST. У будь-якому варіанті зображень, їх образи даються позитивними значеннями пікселів. Тоді узагальнено нормалізацію зображень у вигляді матриць A або векторів a можна представити в стилі NumPy:

$$P = A / np.sum(A); \quad p = a / np.sum(a). \quad (10)$$

В разі перетворення (10) значення елементів матриць A або векторів a будуть знаходитись у діапазоні $[0,1]$, а сума значень вихідних матриць P або векторів p завжди дорівнюватиме одиниці. Тобто, за математичною сутністю образи P та p нічим не відрізняються від розподілень дискретних випадкових величин.

Принципово така нормалізація дозволяє використовувати ти чи інші методи теорії ймовірностей, однак у даній роботі така нормалізація використовується лише для балансу мережі. Нормалізація (10), вочевидь, не займає значного часу і застосовується на виході всіх шарів мережі та на вході нейронів наступних шарів.

Вибір функцій активації дає величезний простір для експериментів дослідників, втім головна вимога: функції активації повинні мати не менше одного вільного параметра. Значення цього параметру або набору

параметрів мають бути оптимізованими при навчанні мережі. У даній роботі в якості такої функції активації обрано параметричний сигмоїд, який визначається формулою:

$$\sigma(\alpha, x) = \frac{1}{1 + \exp[-\alpha(x - 0,5)]} \quad (11)$$

де α – вільний параметр, що підлягає оптимізації; x – значення вхідної величини, наприклад відповідне значення вхідної «ймовірності» p_n для n -го нейрону чергового шару мережі. Графіки параметричних сигмоїдів та їх похідних для деяких значень параметру α представлені на рис. 3.

Як показує аналіз графіків на рис. 3, при $\alpha \rightarrow 0$ функція $\sigma(\alpha, x)$ наближається до лінійної залежності. Тому параметр α можна називати також коефіцієнтом нелінійності.

Зауважимо, що графіки на рис. 3 побудовані для значень аргументу x в діапазоні $[0,1]$. Втім, значення «ймовірностей» p_n можуть бути досить малими при великій кількості нейронів шару. Так, у прикладі структури (1) унормовані значення p_n на вході другого шару матимуть середнє значення $1 / 512 \approx 0,002$. Тобто, значення сигналів на виходах параметричних сигмоїдів всі водночас будуть дуже малими та, що гірше, практично не будуть розрізнятися (див. нижній графік на рис. 3).

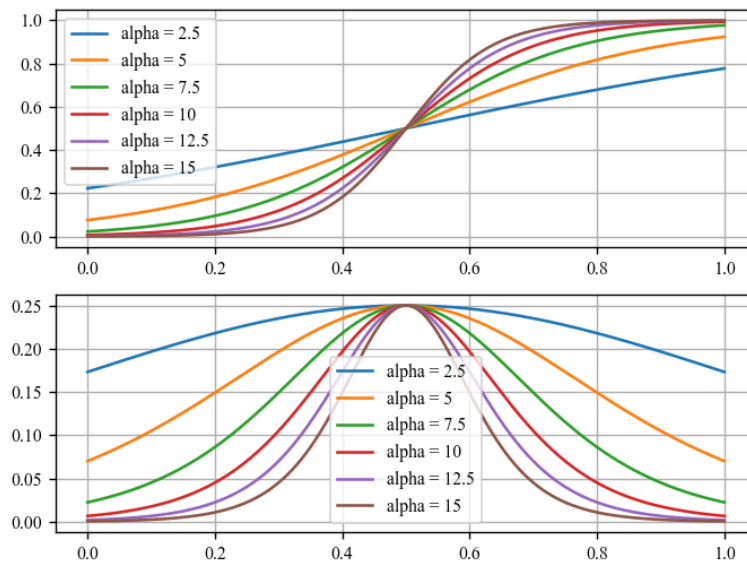


Рис. 3. Графіки параметричних сигмоїдів (зверху) та їх похідних (знизу)

Для того, щоб розмістити вхідні значення у сітці $[0,1]$, виконується дуже швидко та очевидне перетворення масивів значень p_n , яке також простіше всього виразити оператором NumPy: $p^* = p / \text{pr.max}(p)$. Таким чином, основні принципи побудови алгоритмів ПДБ-мереж та умови їх балансу з'ясовано.

Методика проведення експериментів, особливості алгоритмізації та програмування

Згідно звичайного підходу інженерії програмного забезпечення, дослідження здійснювалися за наступними етапами:

1. Аналіз предметної сутності явища (чи об'єкту), що досліджується.
2. Формальна постановка задачі досліджень.
3. Алгоритмізація та програмування.
4. Дослідження та забезпечення якості розроблених програмних продуктів.

1. Аналіз предметної сутності явища (чи об'єкту), що досліджується

Для перевірки працездатності ПДБ-мереж було використано відомий тестовий набір рукописних цифр MNIST. Даний розмічений набір використовуються, наприклад для вирішення навчальних задач за допомогою framework PyTorch Python [9].

На початковому етапі досліджень було виконано візуалізацію частини зразків з MNIST. При цьому багато разів обирались випадковим чином зразки з певними мітками. Візуальний аналіз показав, що ті особи, які, мабуть, заповнювали анкети, не страждали великою любов'ю до каліграфії. Де-які зразки рукописних цифр представлені на рис. 4.

Розглядаючи образи на рис. 4 зліва направо можна задати очевидні питання. Чому другий образ у верхньому рядку розмічений як трійка, а не як сімка? Чому перший образ у другому зверху рядку розмічений як шістка, а не як нуль? Чому перший образ у наступному рядку розмічений як двійка, а не як сімка? Чому четвертий образ у нижньому рядку розмічений як сімка, а не як одиниця? Але саме чудове питання: чому середній образ у третьому зверху рядку розмічений як двійка, а не як конячка, у якої відірвали ноги?

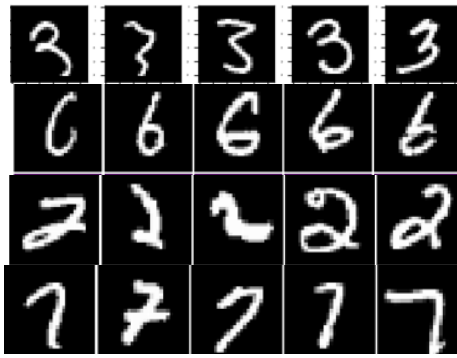


Рис. 4. Зразки рукописних цифр тестового набору MNIST

Втім, наша задача полягає не у навчанні учнів початкової школи з вчителем. Наша задача, все ж таки, полягає у навчанні штучної нейронної мережі, причому без вчителя. Хоча, складності коректного та абсолютно точного навчання очевидні, навіть, з даного експрес-аналізу зображень.

2. Формальна постановка задачі досліджень

У даній роботі вирішується найпростіша задача кластеризації: розміщення потоку вхідних зразків по заданій кількості кластерів. Для тестування алгоритмів обрано частину зразків із навчального набору MNIST, всього 50 000 рукописних цифр.

Кінцева кількість кластерів у цьому простому варіанті обрана як 1000 і задається жорстко. Задача зближення кластерів на даному етапі досліджень не вирішується.

3. Алгоритмізація та програмування

Зазвичай, першими операціями при аналізі зображень є їх нормалізація та, за необхідності, центрування. Наша велика подяка розробникам тестів MNIST: зразки цифр вже представлено у зручному для подальшої обробки вигляді.

В цілому пропонується алгоритм ПДБ-мережі складається з наступних кроків:

- зчитування файлів MNIST та перетворення потоку байтів з цих файлів у числові NumPy-матриці або значення міток;
- попередні перетворення матриць зразків;
- навчання ПДБ-мережі;
- кластеризація без вчителя у навченої ПДБ-мережі;
- валідація результатів кластеризації.

Очевидною особливістю ПДБ-мереж є концентрації впливів близьких нейронів попередніх шарів на близькі за відстанню нейрони наступних шарів. Тому попередні перетворення матриць зразків зводяться до перегруповування пікселів матриць зображень з метою виділення прямокутних фрагментів. Ідея такого перетворення демонструється на рис. 5.

Виконані дослідження показали, що найкращі результати отримуються, якщо матриці MNIST розмірності 28x28 пікселів перенумерувати у підматриці розмірності 7x7 елементів. Тобто умовно матриця зображення поділяється на 16 близьких за топологією груп пікселів. У кожній такій умовній групі, таким чином, 49 елементів.

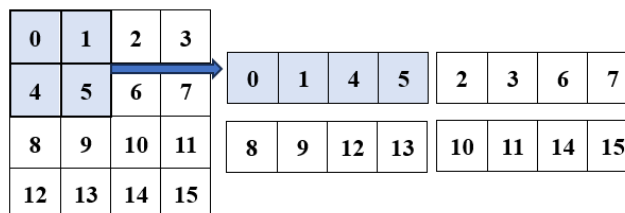


Рис. 5. Перенумерація елементів матриці у номери підматриць

Операція, продемонстрована на рис. 5, віддалено нагадує операції згортки у згорткових мережах. Втім, підматриці визначених розмірів не зміщуються відносно одна одної. Тому дана операція перенумерації виконується ледь не миттєво.

Навчання ПДБ-мережі зводиться до визначення оптимальних значень коефіцієнтів нелінійності параметричних сигмоїдів нейронів. У даній роботі замість навчання тисяч нейронів водночас визначалось єдине значення коефіцієнту нелінійності $\alpha_{optimal}$ водночас для всіх нейронів мережі.

Втім, для успішного вирішення даної задачі, треба визначити досить ефективний показник якості. А для цього треба чітко позначити парадигму тієї самої якості. У даному випадку найбільш бажаний результат полягав би в тому, щоб у визначеній кількості кластерів знайшлось рівно 10 якихось, де була б зосереджена найбільша кількість зразків.

Алгоритм навчання (а потім і кластеризації) передбачає фіксацію кількості зразків, які потрапили у той чи інший кластер. Тому після завершення чергової епохи навчання або кластеризації можна визначити

масив значень `samples` розмірності, що дорівнює кількості кластерів. Тоді характеристики нерівномірності значень цього масиву і можуть бути використані як показники якості кластеризації. Після проведення певних досліджень у даній роботі визначений дуже простий показник нерівномірності: просто дисперсія масиву `samples`. У псевдокоді визначення даного показника можна виразити наступним чином: $D[\text{samples}] = \text{np.var}(\text{samples})$.

Алгоритм навчання зводиться до визначення оптимального значення коефіцієнту нелінійності у циклі на сітці значень параметру `alpha`. Дослідження показали, що залежності $D(\alpha)$ мають полімодальний характер. Тому якісь тонкі методи типу градієнтного спуску, половинного ділення та інші послідовні методи оптимізації у даному дослідженні навіть не використовувались. З урахуванням високої швидкості алгоритмів оптимізація виконувалась методом повного перебору значень параметру `alpha`.

Для навчання при кожному значенні `alpha` випадковим чином обиралась відносно невелика кількість зразків, у нашому випадку 100. Ці зразки й визначали первісні центроїди 100 кластерів. Далі, для заданого значення `alpha` вирішувалась задача кластеризації також невеликого об'єму батча – 10 000 зразків. При цьому для кластеризації використовувався найпростіший метод найближчого сусіда, а центри кластерів зміщувались за допомогою швидких рекурентних перетворень у разі потрапляння чергового зразка у певний кластер.

Дослідження різних залежностей, у тому числі і оптимальних значень параметру нелінійності виконувались для різних структур ПДБ-мереж. У даній роботі представлені варіанти:

`STRUCTURE 1 = ((784, 512), (512, 64))`

`STRUCTURE 2 = ((784, 512), (512, 256), (256, 128), (128, 64))`

`STRUCTURE 3 = ((784, 512), (512, 256), (256, 128), (128, 64), (64, 64))`

Залежності показника оптимальності D від коефіцієнта нелінійності `alpha` для структур 1 та 2 представлені на рис. 6. Як показує аналіз цих графіків, оптимальні значення коефіцієнта нелінійності дуже малі. На наш погляд, це не є випадковістю, оскільки у ПДБ-мережах основні нелінійні перетворення виконують перехідні матриці W .

Після навчання мережі виконують основну функцію кластеризації. Важливим є те, що всі параметри мережі на даному етапі вже визначені. У якості первинних центроїдів для даного прикладу досліджень випадковим чином обираються 1000 зразків. Далі кластеризація відбувається за тим самим алгоритмом, що визначений при розгляді навчання. При цьому значення параметру нелінійності `alpha optimal` вже не корегуються.

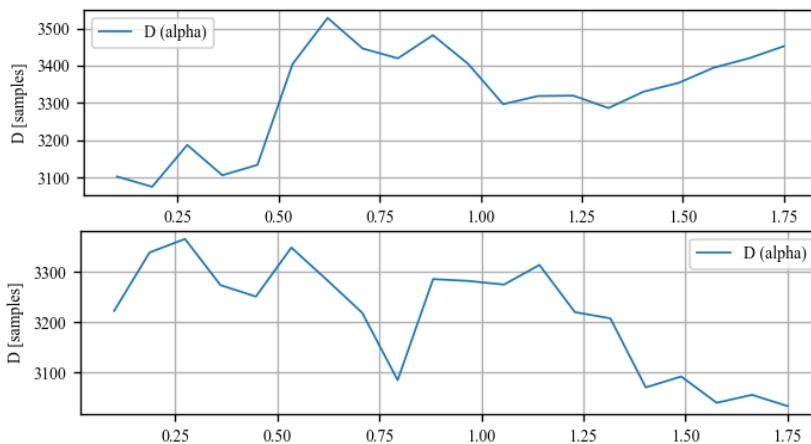


Рис. 6. Залежність дисперсії зразків у кластерах від коефіцієнта нелінійності `alpha` для різних структур ПДБ-мережі (STRUCTURE 1, 2 зверху вниз)

4. Дослідження та забезпечення якості розроблених програмних продуктів

Дослідження якості алгоритмічного та програмного забезпечення здійснювалось методом порівняння результатів кластеризації з різними структурами шарів та шляхом валідації кінцевих рішень.

Для валідації застосована інформація про мітки (labels) зразків рукописних цифр. Дана інформація зберігається у полях об'єктів класу `Cluster`, що показано в ініціалізаторі класу:

```
class Cluster:
    def __init__(self, num_layers, size_last_layer):
        self.num_layers = num_layers
        self.size_last_layer = size_last_layer
        self.center = np.zeros(self.size_last_layer)
        self.num_samples = 0
        self.labels = []
        self.median_label = 0
```


По мірі заповнення кластеру зразками уточнюється їх кількість ($num_samples += 1$), а також до списку міток додається мітка чергового зразка (`labels.append(label)`). **Важливо:** ці мітки ніяким чином не використовуються для вирішення задачі навчання мережі та вирішення задачі кластеризації.

На етапі валідації мітки використовуються **суттєво**. А саме:

- для кожного із 1000 кластерів визначаються медіани списків міток (`median_label`), які приводяться до цілого типу і можуть, таким чином, мати значення лише на множині десятичних цифр: 0, 1, 2, ..., 9;
- сумується кількість зразків (`num_samples`) із кластерів з однаковими медіанами міток;
- також для набору даних MNIST розраховується кількість зразків з однаковими мітками;
- порівнюються обидва ці масиви кількості зразків з однаковими мітками.

Графічно результати валідації для позначених вище структур ПДБ-мереж представлені на рис. 7, а основні числові характеристики результатів – у табл. 1. Аналіз графіків та табличних даних дозволяє зробити такі висновки:

- у всіх випадках повний цикл обробки даних – від завантаження до виводу кінцевих результатів – займає менше 10 хвилин;
- з урахуванням того, що алгоритми та програми виконувались на звичайному комп'ютері та лише із застосуванням обробки на CPU (GPU не використовувалось), такі значення часу рішення можна вважати прийнятними;
- для всіх розглянутих структур максимальне відхилення від даних MNIST не перевищує 10%, а середнє відхилення менше 5%;
- найкращі результати отримано для структури STRUCTURE 2 з чотирма шарами: у даному випадку середнє відхилення становить лише близько 3%, що наближається до результатів вирішення задач класифікації (навчання з вчителем).

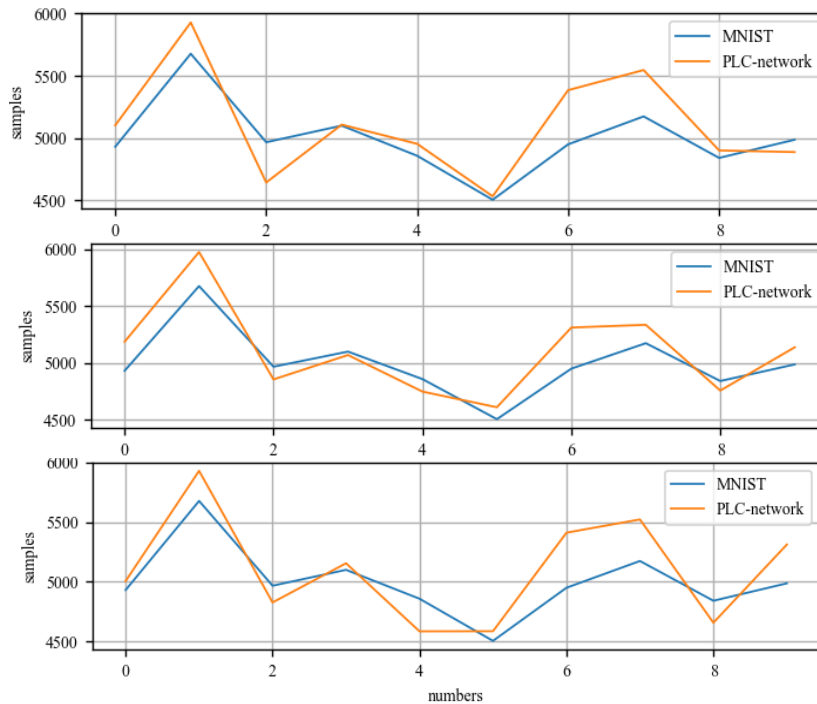


Рис. 7. Результати валідації для різних структур ПДБ-мережі (STRUCTURE 1, 2, 3 зверху вниз)

Таблиця 1

Результати порівняльного аналізу рішень для різних структур ПДБ-мереж

Параметри	Структури		
	STRUCTURE 1	STRUCTURE 2	STRUCTURE 3
Кількість шарів	2	4	5
Alpha optimal	0.62105	0.27368	1.75
R max, %	8.710	7,240	9.22
R mean, %	3.688	3,344	4.388
Час початкових операцій, с	9	9	9
Час навчання, с	209	263	274
Час кластеризації, с	286	288	298
Загальний час вирішення, с	506	563	584

Обговорення, висновки та рекомендації

Пропонований принцип далекоді-близькодії дозволяє суттєво зменшити кількість параметрів штучних нейронних мереж, що потрібно навчати. Так у розглянутому варіанті ПДБ-мережі оптимізується лише один параметр – коефіцієнт нелінійності разом для всіх функцій активації всіх нейронів та всіх шарів мережі.

Первинні гіперпараметри у даному випадку – структура мережі та умовна відстань між шарами мережі. Перехідні матриці (матриці зв'язків між шарами) розраховуються лише один раз на початковому етапі вирішення задачі, що не потребує значного часу.

Перетворення за допомогою таких перехідних матриць за математичною сутністю означає нелінійну дискретну згортку з ядром перетворення у вигляді перехідної матриці W . Тому ПДБ-мережі можна віднести до специфічного класу згорткових нейронних мереж.

У даному дослідженні в найпростішому варіанті ПДБ-мереж отримано прийнятні результати вирішення задачі навчання без вчителя – задачі кластеризації. За показником часу вирішення задач отримано дуже позитивний результат: менше 10 хвилин без використання GPU; за показником точності також прийнятний результат: біля 97% вірних рішень.

Втім, у наступних дослідженнях передбачено виконання порівняльного аналізу з іншими алгоритмічно-програмними засобами за критеріями швидкості та точності.

В цілому, на наш погляд, принцип далекоді-близькодії показав, як мінімум, свою працездатність та доцільність подальшого розвитку відповідної теорії, алгоритмічного та програмного забезпечення.

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. 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.
5. 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.
6. . 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.
7. 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.
8. S. Tiacharoen, "Fuzzy Bat Algorithm for improved Clustering of Convolutional Neural Network Data," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, 2022, pp. 989-992, doi: 10.1109/ITC-CSCC55581.2022.9895077.
9. Welcome to PyTorch Tutorials <https://pytorch.org/tutorials/>. Електронний ресурс] – Режим доступу::

References

1. Odegov M.A. Obgruntuvannya shvidkikh algoritmov 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 "Vimiryuvalna ta obchislyuvalna 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. 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.

5. 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.

6. . 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.

7. 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.

8. S. Tiacharoen, "Fuzzy Bat Algorithm for improved Clustering of Convolutional Neural Network Data," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, 2022, pp. 989-992, doi: 10.1109/ITC-CSCC55581.2022.9895077.

9. Welcome to PyTorch Tutorials<https://pytorch.org/tutorials/> [Електронний ресурс] – Режим доступу:: <https://pytorch.org/tutorials/>