

ДЕРЖАВНИЙ УНІВЕРСИТЕТ "ОДЕСЬКА ПОЛІТЕХНІКА"

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ "ОДЕСЬКА ПОЛІТЕХНІКА"

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

*Кваліфікаційна наукова
праця на правах рукопису*

СМОРОДІН АНДРІЙ ВЯЧЕСЛАВОВИЧ

УДК 004.71

ДИСЕРТАЦІЯ

МЕТОДИ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ
НЕЛІНІЙНОЇ ДИНАМІКИ

Спеціальність: 122 – Комп'ютерні науки

Галузь знань: 12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело



Смородін А.В.

Науковий керівник: **Дмитришин Дмитро Володимирович**, доктор технічних наук, професор

Одеса–2021

АНОТАЦІЯ

Смородін А.В. Методи навчання нейронних мереж на основі нелінійної динаміки. - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки. – Державний університет «Одеська політехніка» МОН України, Одеса, 2021.

У **вступі** обґрунтовано актуальність удосконалення методів навчання нейронних мереж при вирішенні задач класифікації та сегментації об'єктів. Показано можливість скорочення часу навчання нейронних мереж на основі застосування положень нелінійної динаміки при навчанні з використанням алгоритмів градієнтного спуску. Визначено об'єкт, предмет, задачі і методи дослідження; показано зв'язок з науковими програмами та планами; наведено наукову новизну та практичне значення одержаних результатів; висвітлено особистий внесок здобувача.

В **першому розділі** дисертаційної роботи проведено аналіз задач сегментації та класифікації об'єктів на зображеннях із застосуванням нейронних мереж. Показано, що, незважаючи на ефективність існуючих підходів їх вирішення, як правило, побудованих на глибоких нейронних мережах (ГНМ) складної архітектури, існуючі тенденції великих даних (великі обсяги, різноманітність, мінливість) призводять до закономірної проблеми – збільшення обчислювальних ресурсів, які витрачаються на їх навчання та перенавчання. За останні 10 років, за відомостями компанії OpenAI спостерігається збільшення в 300 тис. разів витрачених на навчання ГНМ обчислювальних ресурсів. Аналіз показав, що час навчання ГНМ значною мірою визначається оптимізаційними методами, які використовуються. Тому, важливою науково-практичною задачею, яка вирішується у дисертаційній роботі, є скорочення часу навчання нейронних мереж при вирішенні задач сегмен-

тації та класифікації об'єктів шляхом удосконалення оптимізаційних методів, які для цього використовуються.

На основі аналізу існуючих методів навчання ГНМ, які входять в сучасні бібліотеки розробки нейронних мереж, виявлено, що більшість із них засновані на методі градієнтного спуску, який в міні-пакетному режимі навчання формально виглядає наступним чином:

$$\Theta_{t+1} = \Theta_t - \frac{1}{B} \gamma \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta_t), \bar{y}_i) \quad (1)$$

де Θ_t – параметри нейронної мережі на момент часу t ; B – кількість елементів в одному міні-пакеті тренувальних даних; $L(f(\bar{x}_i; \Theta_t), \bar{y}_i)$ – функція втрат, що відображає похибку між передбаченими значеннями $f(\bar{x}_i; \Theta_t)$ та очікуваними результатами $\bar{y}_i$; $f(\bar{x}_i; \Theta_t)$ – функція, що позначає відповідні перетворення, які виконуються нейронною мережею з внутрішніми параметрами Θ_t та вхідними параметрами $\bar{x}_i$; γ – коефіцієнт навчання.

Якщо ввести додаткову функцію від параметра ϑ :

$$G(\vartheta) = \vartheta - \frac{1}{B} \gamma \nabla_{\vartheta} \sum_{i=1}^B L(f(\bar{x}_i; \vartheta), \bar{y}_i),$$

то вираз (1) можна представити у вигляді класичної дискретної динамічної системи (ДДС).

$$\Theta_{t+1} = G(\Theta_t). \quad (2)$$

Таким чином, вираз (2) дозволяє представити метод градієнтного спуску у вигляді ДДС, до якої можна застосувати положення нелінійної динаміки. Показано, що таке представлення дозволяє розробляти модифікації ДДС (2) з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку.

З урахуванням (1) та (2) запропоновано модифікацію методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням в ДДС:

$$\Theta_{t+1} = (1 - \lambda)G \left(\sum_{j=1}^N a_j \Theta_{t-jT+T} \right) + \lambda \sum_{j=1}^N b_j \Theta_{t-jT+1}. \quad (3)$$

У (3) параметри a_j та b_j повинні задовольняти умовам $\sum_j a_j = \sum_j b_j = 1, \forall \lambda \in \mathbb{R}$.

Для їх обчислення необхідна розробка відповідного математичного апарату.

З урахуванням (1) і (2) запропоновано модифікацію методу градієнтного спуску з урахуванням прогнозуючого зворотного зв'язку в ДДС:

$$\Theta_{t+1} = G(\Theta_t) + u_t, \quad (4)$$

де прогнозуючий зворотній зв'язок $u_t = (a_1 - 1)G(\Theta_t) + \sum_{j=2}^N a_j G^{((j-1)T+1)}(\Theta_t)$,

$G^{(1)}(\Theta) = G(\Theta)$, $G^{(k)}(\Theta) = G(G^{(k-1)}(\Theta))$, $\sum_{j=1}^N a_j = 1$, $k \in \mathbb{N}$.

Таким чином, удосконалено метод градієнтного спуску на основі положень нелінійної динаміки (**перший пункт новизни наукового дослідження**), що дозволило розробити модифікації методу з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС та застосувати його як при навчанні нейронних мереж з метою зменшення часу навчання так і для вирішення інших оптимізаційних задач.

В другому розділі розроблено метод пошуку параметрів для запропонованих модифікацій методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС.

Запропонований метод пошуку параметрів реалізується наступною послідовністю:

Етап 1. Рішення оптимізаційної задачі на класі комплексних поліномів.

При вирішенні оптимізаційної задачі параметри комплексних поліномів необхідно вибирати таким чином, щоб множина $(\bar{\mathbb{C}} \setminus \Phi(\hat{\mathbb{D}}))^*$ на комплексній площині була найбільш розтягнутою в потрібному напрямку, наприклад, у від'ємному напрямку дійсної осі. Така задача розглядається на класі однолистных функцій та типово-дійсних функцій по Рогозинському. Застосовані означення: $\bar{\mathbb{C}}$ – розширена на комплексна площина, $\hat{\mathbb{D}}$ – замкнутий центральний одиничний круг, зірочкою позначена операція інверсії $(z)^* = \frac{1}{z}$, а функція $\Phi(z) = (1 - \lambda)^T \frac{z \left(\sum_{j=0}^{N-1} a_j z^j \right)^T}{\left(1 - \lambda \sum_{j=1}^N b_j z^j \right)^T}$.

Рішення зводиться до розв'язання екстремальних задач, для яких знаходяться екстремальні значення, та екстремайзери.

1. Для полінома $F(z) = \sum_{j=1}^N a_j z^j$ з дійсними коефіцієнтами та schlicht- нормуванням $a_1 = 1$. Знайти $\sup_{a_2, \dots, a_N} \left(\inf_{z \in \mathbb{D}} \{ \Re(F(z)) : \Im(F(z)) = 0 \} \right)$.

Екстремальне значення є таким $-\frac{1}{4} \sec^2 \frac{\pi}{N+2}$, а відповідний екстремайзер

$$F(z) = \frac{1}{U'_N \left(\cos \frac{\pi}{N+2} \right)} \sum_{j=1}^N U'_{N-j+1} \left(\cos \frac{\pi}{N+2} \right) U_{j-1} \left(\cos \frac{\pi}{N+2} \right) z^j$$

є єдиним та однолистным, $U_j(x)$ - поліноми Чебишова другого роду.

2. У класі T_N усіх типово дійсних поліномів ступеню N schlicht нормуванням $a_1 = 1$ знайти величину $J_N = \max_{p(z) \in T_N} \sup_{z \in \mathbb{D}} \{ |F_n(z)| \}$. Екстремальне значення

$$\text{визначено як } J_N = \begin{cases} \frac{1}{4 \sin^2 \frac{\pi}{2(N+2)}}, & N - \text{непарне} \\ \frac{1}{4 \sin^2 \vartheta}, & N - \text{парне} \end{cases}, \text{ де } \vartheta \text{ мінімальний позитив-}$$

ний корінь рівняння $(N+3) \cos(N+1)\vartheta + (N+1) \cos(N+3)\vartheta = 0$. Наведено алгоритм обчислення коефіцієнтів екстремайзера.

3. У класі однолистных з T -круговою симетрією поліномів $F_N^{(T)} =$

$= z \left(1 + \sum_{j=2}^N a_j z^{(j-1)T} \right)$ знайти асимптотичну оцінку величини

$\max_{z \in \mathbb{D}} |F_N^{(T)}(z)|$. Екстремальне асимптотичне значення при $N \rightarrow \infty$, $\max_{z \in \mathbb{D}} |F_N^{(T)}| \sim \frac{1}{2^{2/T} c_T} N^{2/T}$, де $c_T = \pi^{\frac{2}{T}-1} \Gamma^2 \left(\frac{1}{T} + \frac{1}{2} \right)$, $\Gamma(\xi)$ - гама-функція.

Екстремайзером $F^{(0)}(z) = z + \sum_{j=2}^N a_j^{(0)} z^{T(j-1)+1}$, є

$$a_j^{(0)} = \left(1 - \frac{(j-1)T}{1 + (N-1)T} \right) \prod_{k=1}^{j-1} \frac{\sin \frac{\pi(2+T(j-1))}{2+T(N-1)}}{\sin \frac{\pi T j}{2+T(N-1)}}, j = 2, \dots, N.$$

Коефіцієнти екстремайзерів, які екстремально розтягують або стискають центральний одиничний круг на комплексній площині, використовуються для чисельного моделювання з метою визначення допустимих параметрів.

Етап 2. Проведення чисельного моделювання з використанням варіацій допустимих параметрів, з метою визначення параметрів a_j , b_j та λ удосконаленого методу градієнтного спуску (3), (4).

Етап 3. Експериментальна перевірка знайдених параметрів для різних завдань.

Розроблений метод пошуку параметрів для запропонованих модифікацій методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС на основі розв'язання оптимізаційних задач на класі комплексних поліномів, який дозволяє забезпечити їх реалізацію для широкого кола оптимізаційних задач, є **другим пунктом наукової новизни**.

У **третьому розділі** набули подальшого розвитку методи навчання нейронних мереж шляхом запропонованих модифікацій методу градієнтного спуску. Проведено їх дослідження для різних архітектур нейронних мереж.

Розроблено на основі (3) метод навчання нейронних мереж шляхом модифікації методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням

в ДДС при заданих параметрах $T = 1, N = 2$:

$$\Theta_{t+1} = ((1 - \lambda)a + \lambda b)\Theta_t + (1 - ((1 - \lambda)a + \lambda b))\Theta_{t-1} - (1 - \lambda)\frac{1}{B}\gamma\nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; a\Theta_t + (1 - a)\Theta_{t-1}), \bar{y}_i), \quad (5)$$

де параметри a, b и λ визначаються у відповідності з наведеним вище методом пошуку параметрів.

Розроблено на основі (4) метод навчання нейронних мереж шляхом модифікації методу градієнтного спуску з урахуванням прогнозуючого зворотного зв'язку в ДДС при заданих параметрах $T = 1, N = 2$:

$$\begin{cases} y_t = G(\Theta_t) \\ x_{t+1} = G(a + (1 - a)y_t) \end{cases} \quad (6)$$

або

$$\begin{cases} y_t = G(\Theta_t) \\ x_{t+1} = aG(\Theta_t) + (1 - a)G(y_t) \end{cases} \quad (7)$$

Ці схеми було покладено в основу створення низки нових схем реалізації методу градієнтного спуску. Проведено їх тестування на прикладах.

1. Розв'язана оптимізаційна задача $\min_{x \in \mathbb{R}} \max_{y \in \mathbb{R}} [\Phi(x, y)]$ для функції $\Phi(x, y) = 2x^2 + y^2 + 4xy + \frac{4}{3}y^2 - \frac{1}{4}y^4$, для яких відомі 3 критичні точки (точка z_1 - точка локального мінімуму, а точки z_0 и z_2 - сідлові точки) в яких $\nabla\Phi(x, y) = 0$, а саме $z_0 = (0, 0)$; $z_1 = (-2 - \sqrt{2}, 2 + \sqrt{2})$; $z_2 = (-2 + \sqrt{2}, 2 - \sqrt{2})$.

Проведено експерименти з побудови басейнів тяжіння до екстремальних то-

чок. Обчислення проводилися для невеликих областей локалізації екстремальних точок з накладеною на них дрібною сіткою за допомогою розробленої на мові програмування GoLang багатопотокової програми. Результати аналізу показали, що модифікації (6) і (7) призводять до збіжності до всіх трьох точок, включаючи сідлові (рис. 1), та потребують менш обчислювальних ресурсів, ніж методи оптимізації, які вимагають обчислення Гессіану. Здатність збіжності до сідлових точок є ключовою перевагою, завдяки чому можливо значно поліпшити навчання GAN мереж (generative adversarial networks).

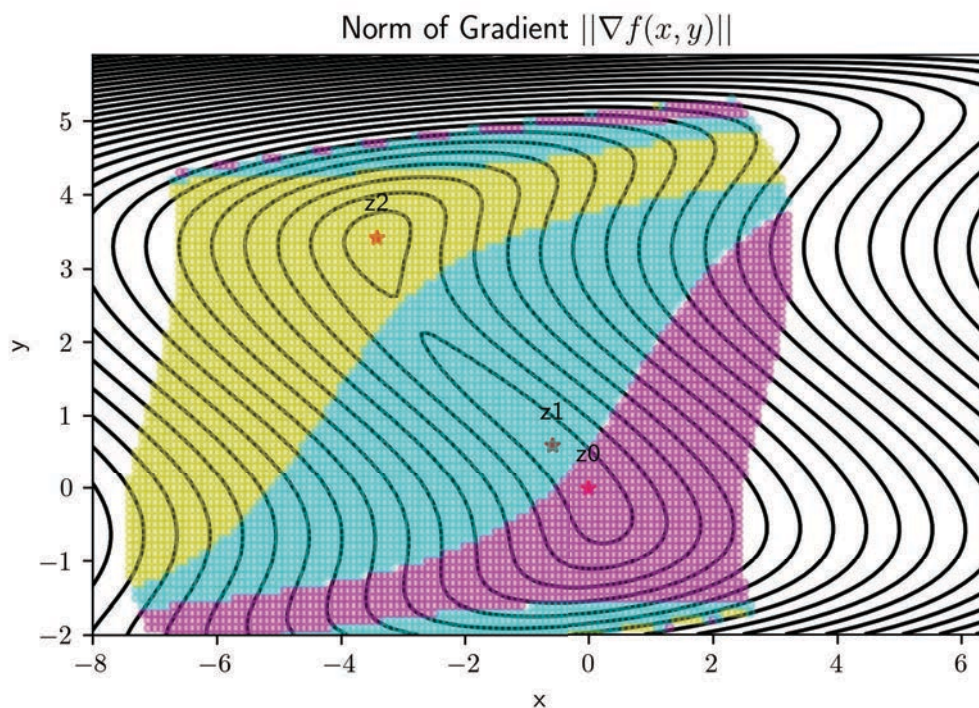


Рисунок 1: Контурний графік $\|\Phi(x, y)\|$ (блакитним кольором позначено басейн тяжіння до точки z_1 , жовтим до z_2 , пурпурним до z_0)

2. Розв'язана оптимізаційна задача для типових тестових функцій, однією з яких є функція Розенброка $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$, яка має глобальний мінімум у точці $(x, y) = (1, 1)$.

Проведено порівняльний аналіз результатів пошуку екстремуму функції Розенброка, яку було надано у вигляді нейронної мережі, досліджуванням та найпопулярнішими методами для навчання глибоких нейронних мереж (SGD (Stochastic Gradient Decent) алгоритмом з моментом та методом Нестерова, алгоритмом Adam) з використанням бібліотеки PyTorch компанії Facebook, що містить градієнтні алгоритми оптимізації. Численні експерименти показали перевагу розробленого методу (New SGD algorithm with control) як у швидкості збіжності, так і в більш плавній поведінці функції втрат (рис. 2).

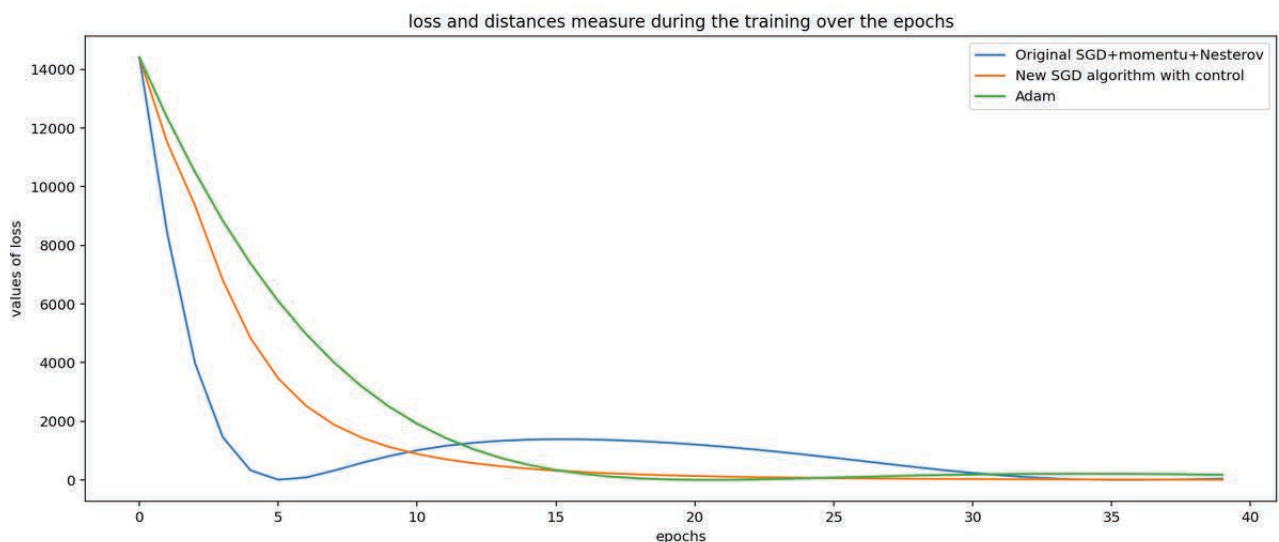


Рисунок 2: Графік функції втрат при пошуку екстремуму функції

3. Навчання багатошарового перцептрона для розпізнавання рукописних цифр.

В якості навчального та тестового наборів даних був використаний MNIST (60,000 зображень помічених вручну). Аналіз функції помилки навчання багатошарового перцептрона при розпізнаванні рукописних цифр показав, що розроблений метод забезпечує скорочення часу навчання (до трьох разів) при меншому рівні помилки (рис. 3).

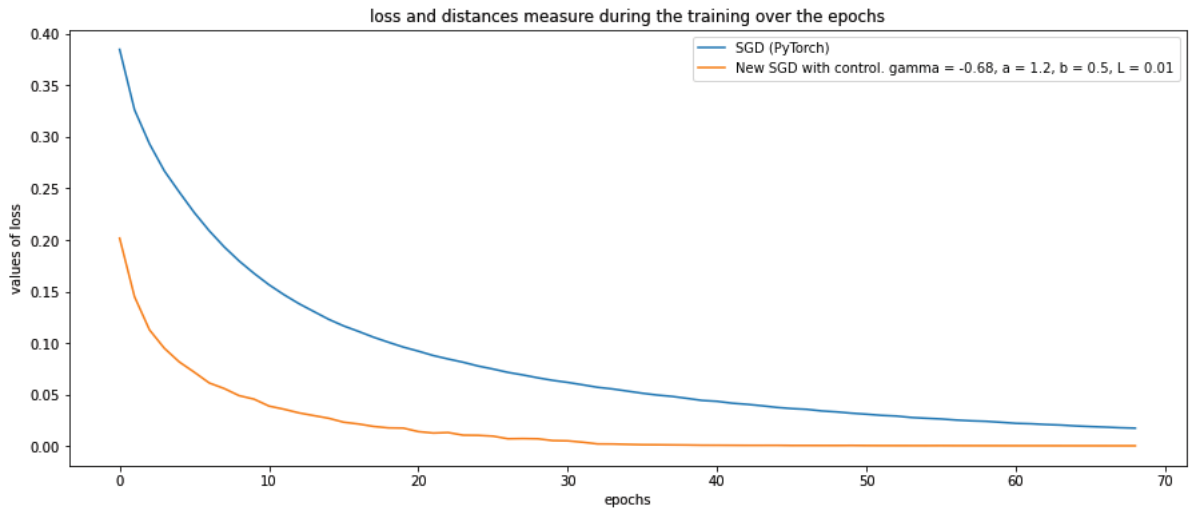


Рисунок 3: Графік функції помилки навчання розпізнавання рукописних цифр

Таким чином, набули подальшого розвитку методи навчання нейронних мереж на основі запропонованих модифікацій методу градієнтного спуску, що дозволило зменшити час навчання нейронних мереж різних архітектур при вирішенні задач сегментації та класифікації об'єктів (**третій пункт новизни наукового дослідження**).

У **четвертому розділі** розроблена програмна реалізація запропонованих методів навчання нейронних мереж на основі нелінійної динаміки та проведено їх апробацію при побудові підсистеми сегментації та класифікації ортопантомограм для стоматологічної інтелектуальної інформаційної системи.

При побудові підсистеми вирішувалась задача перетворення ортопантомограм в *одонтограми*, які є спеціалізованою формою представлення інформації лікаря про стан кожного зуба пацієнта, з застосуванням згорткової нейронної мережі.

На першому етапі відбувалось формування навчальної вибірки. Оскільки створення необхідної навчальної вибірки є неможливим у зв'язку з високою ціною отримання панорамних зображень, необхідністю їх фільтрації від сторонніх

шумів і складністю проведення початкової ручної сегментації стоматологами, використано метод аугментації навчальної вибірки з використанням як геометричних, так і піксельних перетворень. Це дозволило сформувати 300 тис. зображень на основі початкового набору з 53 зображень і 32 бінарних масок. Всі зображення були збережені як багатовимірний тензор PyTorch в спеціальному внутрішньому форматі.

Для вирішення задачі сегментації обрано архітектуру гібридної згорткової нейронної мережі на основі попередньо навченого мережевого кодера ResNet-18 і декодера з мережі U-Net. Навчання цієї мережі передбачає налаштування близько 18 мільйонів параметрів. В якості функції помилки запропоновано використовувати лінійну комбінацію крос-ентропії і коефіцієнта Дайса, що добре зарекомендувала себе при навчанні згорткових нейронних мереж. Навчання проводилося з використанням бібліотеки PyTorch з різними модифікаціями методу градієнтного спуску. Досліджено використання оперативної пам'яті GPU, мінімальне значення функції втрати (Мінімум) і значення функції втрати (Тест) на тестовому наборі даних (таблиця 1).

Таблиця 1: Порівняльний аналіз алгоритмів навчання

Назва	GPU пам'ять (MiB)	Мінімум	Тест
RMSProps	3503	1.3766	0.5385
Adagrad	3571	0.0502	0.5269
Adadellta	3459	0.5520	0.5560
Adam	3459	0.0474	1.189
SGD	3450	0.2197	0.5365
New method (3)	3471	0.2673	0.2049

Аналіз показав, що запропонований метод, за помилкою навчання має аналогічні результати на навчальній вибірці, а на тестовій спостерігається її зменшення. Крім того при однакових витратах оперативної пам'яті GPU для розроблених методів функція втрати в два рази менше на тестових даних в порівнянні з SGD і

майже в 7 разів в порівнянні з алгоритмом Adam ([18]).

Графіки функції втрати під час навчання нейронної мережі (рис. 4), показали, що New method (3) майже вдвічі швидше мінімізує функцію втрати в порівнянні з SGD.

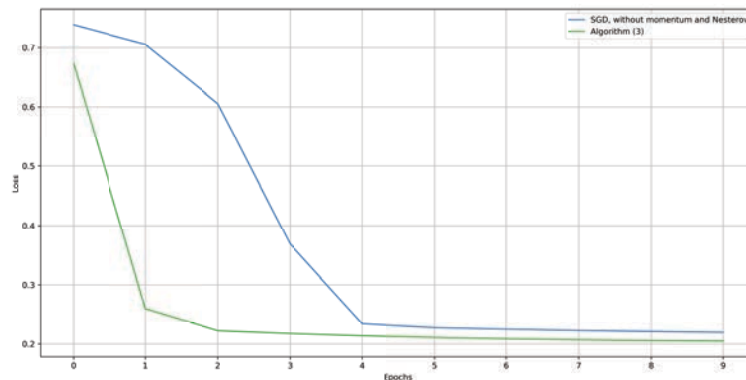


Рисунок 4: Графік порівняння швидкості мінімізації функції втрати алгоритмів SGD (синій) та методу (3) (зелений) від часу навчання

У **висновках** сформульовано отримані в ході роботи над дисертацією науково-практичні результати.

Ключові слова: методи навчання нейронних мереж, оптимізація, згорткові нейронні мережі, сегментація, класифікація.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Основні матеріали дисертаційної роботи і головні результати проведених досліджень були опубліковані в сьоми наукових роботах (4 статті – у науково-метричних базах Scopus, 3 першого квартиля Q1; 3 – у фахових збірниках, 3 без співавторів), а також у одинадцяті апробаційних матеріалах.

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Dmitrishin D., Smorodin A., Stokolos A. On a family of extremal polynomials. *Comptes Rendus Mathematique*. 2019. 357(7). P. 591-596. DOI:<https://doi.org/10.1016/j.crma.2019.06.010>. (Scopus, Q1).
2. Dmitrishin D., Smorodin A., Stokolos A., Tohaneanu M. Symmetrization of Suffridge polynomials and approximation of T-symmetric Koebe functions. *Journal of Mathematical Analysis and Applications*. 2021. 503(3). DOI:<https://doi.org/10.1016/j.jmaa.2021.125350>. (Scopus, Q1).
3. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. *Applied and Computational Harmonic Analysis*. 2021. 56. P. 283-305. DOI:<https://doi.org/10.1016/j.acha.2021.08.008>. (Scopus, Q1)
4. Smorodin A. Predictive control methods in tasks of searching saddle points. *Proceedings of Odessa Polytechnic University*. 2020. V. 3(62). P. 80-90. DOI:10.15276/opu.3.62.2020.10. URL:<https://pratsi.op.edu.ua/articles/show/20968>
5. Smorodin A. The use of control theory methods in training neural networks on the example of teeth recognition on panoramic x-ray images//Automation of Technological and Business Processes. 2021. V. 13(2). P. 36-40. URL:<http://dx.doi.org/10.15673/atbp.v13i2.2055>
6. Smorodin, A.V. Використання методів теорії управління при вивченні нейронних мереж на прикладі рукописного тексту. Прикладні аспекти інформаційних технологій. V. 4(3). 2021. P. 243–249. URL:<http://aait.ccs.od.ua/index.php/journal/article/view/119>

7. Dillies, J., Dmitrishin, D., Smorodin, A., Stokolos A. (2022). On the Koebe Quarter Theorem for Polynomials. Proceedings of the International Geometry Center, V. 14(3). 2022. P. 219-230. <https://doi.org/10.15673/tmgc.v14i3.2057>. (Scopus, Q4)

Наукові праці, які засвідчують апробацію матеріалів дисертації

8. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. Harmonic Analysis and Partial Differential Equations. Canadian Mathematical Society. 2021.
9. Dmitrishin D., Smorodin A., Stokolos A. Extremal polynomials and geometric complex analysis. Algebraic and Geometric Methods of Analysis. International Online Conference. 2020.
10. Smorodin A. On question by Dimitrov. Analysis Seminar. Odessa. I. I. Mechnikov National University. 2018.
11. Смородин А. Методы прогнозирующего управления в задачах поиска седловых точек и обучения нейронных сетей. /Актуальні питання фізико-математичних та технічних наук: теоретичні та прикладні дослідження. 2021. Р. 13-23. II Міжнародна науково-практична інтернет-конференція. URL: https://openscilab.org/wp-content/uploads/2021/05/aktualni-pitanja-fiziko-matematichnih-ta-tehnichnih-nauk-teoretichni-ta-prikladni-doslidzhennja_2021_05_21.pdf
12. Смородин А. Методи прогнозного управління для задач оптимізації. Теоретико-практичні проблеми використання математичних методів та комп'ютерно-

орієнтованих технологій в освіті та науці. 2021. III Всеукраїнська науково-практична онлайн-конференція.

13. Dmitrishin D., Stokolos A., Smorodin A. On a new family of extremal polynomials. The 35th annual Southeastern Analysis. Meeting, University of Alabama, 2019.
14. Dmitrishin D., Stokolos A., Smorodin A. An extremal problem for polynomials. CMS 75th +1 Anniversary Summer Meeting, Montreal, 2021
15. Dmitrishin D., Skrinnik I., Smorodin A., Stokolos A. Dimitrov's question for the polynomials of degree 1,2,3,4,5,6. 2018. arXiv:1808.08636. URL: <https://arxiv.org/abs/1808.08636>.
16. Dmitrishin D., Smorodin A., Stokolos A. Estimating the Koebe radius for polynomials. 2018. arXiv: 1805.06927. URL: <https://arxiv.org/abs/1805.06927v1>
17. Dmitrishin D., Smorodin A., Stokolos. On C. Michel's hypothesis about the modulus of typically real polynomials. 2020. arXiv: 2005.12432. URL: <https://arxiv.org/abs/2005.12432>
18. Dillies J., Dmitrishin D., Smorodin A., Stokolos. On the Koebe Quarter Theorem for Polynomials. 2019. arXiv: 1904.11039. URL: <https://arxiv.org/abs/1904.11039>

ABSTRACT

Smorodin A.V. Methods of neural networks training on the basis of nonlinear dynamics. – Qualification scientific work on the rights of the manuscript.

PhD degree dissertation – Doctor of Philosophy in specialty 122 – Computer Science.– Odessa Polytechnic State University Ministry of Education and Science of Ukraine of Ukraine, Odessa, 2021.

In the **introduction** the relevance of improving the methods of neural networks training to solve the problems of classification and segmentation is proved. The possibility of reducing the training time of neural networks is shown based on the application of provisions of nonlinear dynamics when using gradient descent algorithms. Object, subject, tasks and methods of research are determined; the connection with scientific programs and plans is shown; scientific novelty and practical significance of the obtained results are given; the applicant's personal contribution is highlighted.

The **first section** of the dissertation analyzes the problems of segmentation and classification of objects in images using neural networks. It is shown that despite the effectiveness of existing approaches to their solution usually based on deep neural networks (DNN) of complex architecture, existing trends of big data (large volumes, diversity, variability) lead to a natural problem – an increase in computing resources spent on their training and retraining (over the past 10 years OpenAI has seen a 300,000-fold increase in computing resources spent on DNN training). The analysis showed that the training time of DNN is largely determined by the optimization methods used. Therefore, the important scientific and practical task, which is solved in this dissertation, is to reduce the training time of neural networks in solving the problems of segmentation and classification by improving the optimization methods used for this purpose.

Based on the analysis of existing methods of training DNN which are included

in modern libraries of neural networks development, it was found that most of them are based on the method of gradient descent, which in the mini-batch learning mode looks like this:

$$\Theta_{t+1} = \Theta_t - \frac{1}{B} \gamma \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta_t), \bar{y}_i) \quad (1)$$

where Θ_t – neural network parameters at the time t , B – number of items in one mini training data package, $L(f(\bar{x}_i; \Theta_t), \bar{y}_i)$ – loss function, which reflects the error between the predicted values $f(\bar{x}_i; \Theta_t)$ and expected results $\bar{y}_i$, $f(\bar{x}_i; \Theta_t)$ – function that denotes the corresponding transformations performed by the neural network with internal parameters Θ_t and input parameters learning $\bar{x}_i$; γ – learning rate.

If one enters an additional function from the parameter ϑ :

$$G(\vartheta) = \vartheta - \frac{1}{B} \gamma \nabla_{\vartheta} \sum_{i=1}^B L(f(\bar{x}_i; \vartheta), \bar{y}_i)$$

then expression (1) can be represented as the classic discrete dynamic system (DDS).

$$\Theta_{t+1} = G(\Theta_t). \quad (2)$$

Thus, the expression (2) allows to represent a method of gradient descent in the form of the DDS, to which the position of nonlinear dynamics can be applied. It is shown that such presentation allows to develop modifications of the DDS (2) considering the delayed and predictive feedback.

Taking into account (1) and (2), a modification of the gradient descent method is proposed considering the delayed feedback in the DDS:

$$\Theta_{t+1} = (1 - \lambda)G \left(\sum_{j=1}^N a_j \Theta_{t-jT+T} \right) + \lambda \sum_{j=1}^N b_j \Theta_{t-jT+1}. \quad (3)$$

In (3) parameters a_j and b_j must meet the conditions $\sum_j a_j = \sum_j b_j = 1, \forall \lambda \in \mathfrak{R}$.

To calculate them, it is necessary to develop a suitable mathematical apparatus.

Taking into account (1) and (2), a modification of the gradient descent method is proposed considering the predictive feedback in the DDS:

$$\Theta_{t+1} = G(\Theta_t) + u_t, \quad (4)$$

where the predictive feedback $u_t = (a_1 - 1)G(\Theta_t) + \sum_{j=2}^N a_j G^{((j-1)T+1)}(\Theta_t)$, $G^{(1)}(\Theta) = G(\Theta)$, $G^{(k)}(\Theta) = G(G^{(k-1)}(\Theta))$; $\sum_{j=1}^N a_j = 1, k \in \mathbb{N}$.

Thus, the method of gradient descent based on nonlinear dynamics (**the first point of novelty of the scientific research**) was improved which allowed to develop modifications of the method considering the delayed and predictive feedback in the DDS and apply it in neural networks training in order to reduce the training time and to solve other optimization problems.

In the second section the method of search of parameters for the offered modifications of the gradient descent method considering the delayed and predictive feedback in the DDS is developed.

The proposed method of searching for parameters is implemented in the following sequence:

Stage 1. Solution of the optimization problem on the class of complex polynomials.

When solving the optimization problem, the parameters of complex polynomials must be selected so that the set $(\bar{\mathbb{C}} \setminus \Phi(\bar{\mathbb{D}}))^*$ on the complex plane was the most

stretched in the desired direction, for example, in the negative direction of the real axis. Such problem is considered in the class of univalent and typically-real Rogozinski functions. Applied definitions: $\bar{\mathbb{C}}$ - extended complex plane, $\hat{\mathbb{D}}$ - closed central unit circle, an asterisk indicates an inversion operation $(z)^* = \frac{1}{z}$, and the function

$$\Phi(z) = (1 - \lambda)^T \frac{z \left(\sum_{j=0}^{N-1} a_j z^j \right)^T}{\left(1 - \lambda \sum_{j=1}^N b_j z^j \right)^T}.$$

The solution comes down to solving optimization problems for which extreme values and extremizers were found.

1. For polynomial $F(z) = \sum_{j=1}^N a_j z^j$ with valid coefficients and schlicht-rationing $a_1 = 1$. Find $\sup_{a_2, \dots, a_N} \left(\inf_{z \in \mathbb{D}} \{ \Re(F(z)) : \Im(F(z)) = 0 \} \right)$.

The extreme value is $-\frac{1}{4} \sec^2 \frac{\pi}{N+2}$, and the corresponding extremizer

$$F(z) = \frac{1}{U'_N \left(\cos \frac{\pi}{N+2} \right)} \sum_{j=1}^N U'_{N-j+1} \left(\cos \frac{\pi}{N+2} \right) U_{j-1} \left(\cos \frac{\pi}{N+2} \right) z^j$$

is single and univalent, $U_j(x)$ - Chebyshev polynomial of the second kind.

2. In class T_N ll typically valid polynomials of degree N with schlicht-rationing

$a_1 = 1$ find value $J_N = \max_{p(z) \in T_N} \sup_{z \in \mathbb{D}} \{ |F_n(z)| \}$. Extreme value is defined as $J_N =$

$$\begin{cases} \frac{1}{4 \sin^2 \frac{\pi}{2(N+2)}}, & N - \text{odd} \\ \frac{1}{4 \sin^2 \vartheta}, & N - \text{even} \end{cases}, \text{ where } \vartheta \text{ minimum positive root of the equation } (N +$$

3) $\cos(N+1)\vartheta + (N+1)\cos(N+3)\vartheta = 0$. The algorithm for calculating extremizer coefficients is given.

3. In the class of univalent T -fold symmetry polynomials $F_N^{(T)} =$

$$= z \left(1 + \sum_{j=2}^N a_j z^{(j-1)T} \right) \text{ find an asymptotic estimate of the value}$$

$\max_{z \in \mathbb{D}} |F_N^{(T)}(z)|$. Extreme asymptotic value at $N \rightarrow \infty$, $\max_{z \in \mathbb{D}} |F_N^{(T)}| \sim \frac{1}{2^{2/T} c_T} N^{2/T}$, where $c_T = \pi^{\frac{2}{T}-1} \Gamma^2\left(\frac{1}{T} + \frac{1}{2}\right)$, $\Gamma(\xi)$ - gamma-function.

Extremizer $F^{(0)}(z) = z + \sum_{j=2}^N a_j^{(0)} z^{T(j-1)+1}$, is

$$a_j^{(0)} = \left(1 - \frac{(j-1)T}{1 + (N-1)T}\right) \prod_{k=1}^{j-1} \frac{\sin \frac{\pi(2+T(j-1))}{2+T(N-1)}}{\sin \frac{\pi T j}{2+T(N-1)}}, j = 2, \dots, N.$$

The extremizer coefficients that extremely stretch or compress the central unit circle on the complex plane are used for numerical simulations to determine the allowable parameters.

Stage 2. Carrying out of numerical simulations using variations of the allowable parameters to determine the parameters a_j , b_j and λ of the improved gradient descent method (3), (4).

Stage 3. Experimental verification of the parameters found for various problems.

The method of searching for parameters for the proposed modifications of the gradient descent method considering the delays and predictive feedback in the DDS based on the solution of optimization problems in the class of complex polynomials, which allows their implementation for a wide range of optimization problems, is **the second point of scientific novelty**.

In **the third section** methods of neural network training were further developed based on the proposed modifications of the gradient descent method. The research for different architectures of neural networks was carried out.

The method of neural networks training based on the modification of the gradient descent method considering the delayed feedback in the DDS is developed on the basis

of (3) at the specified parameters $T = 1, N = 2$:

$$\begin{aligned} \Theta_{t+1} = & ((1 - \lambda)a + \lambda b)\Theta_t + (1 - ((1 - \lambda)a + \lambda b))\Theta_{t-1} \\ & - (1 - \lambda)\frac{1}{B}\gamma\nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; a\Theta_t + (1 - a)\Theta_{t-1}), \bar{y}_i), \end{aligned} \quad (5)$$

where parameters a, b and λ are determined in accordance with the above method of searching for parameters.

The method of neural networks training based on the modification of the gradient descent method considering the predictive feedback in the DDS is developed on the basis of (4) at the specified parameters $T = 1, N = 2$:

$$\begin{cases} y_t = G(\Theta_t) \\ x_{t+1} = G(a + (1 - a)y_t) \end{cases} \quad (6)$$

or

$$\begin{cases} y_t = G(\Theta_t) \\ x_{t+1} = aG(\Theta_t) + (1 - a)G(y_t) \end{cases} . \quad (7)$$

These schemes were the basis for the creation of a number of new schemes for the implementation of the gradient method descent. They were tested on examples.

1. The optimization problem $\min_{x \in \mathbb{R}} \max_{y \in \mathbb{R}} [\Phi(x, y)]$ for function $\Phi(x, y) = 2x^2 + y^2 + 4xy + \frac{4}{3}y^2 - \frac{1}{4}y^4$, for which 3 critical points are known (point z_1 - point of the local minimum, and the points z_0 and z_2 - saddle points) in which $\nabla \Phi(x, y) = 0$, namely $z_0 = (0, 0)$; $z_1 = (-2 - \sqrt{2}, 2 + \sqrt{2})$; $z_2 = (-2 + \sqrt{2}, 2 - \sqrt{2})$.

Experiments were carried out to build pools of attraction to extreme points. The calculations were performed for small areas of localization of extreme points with

a fine grid superimposed on them using a multi-threaded program developed in the GoLang programming language. The results of the analysis showed that modifications (6) and (7) lead to convergence to all three points, including saddles (Fig. 1), and require less computational resources than optimization methods that require the calculation of Hessian. The ability to convergence to saddle points is the key advantage making it possible to significantly improve the training of GAN networks (generative adversarial networks).

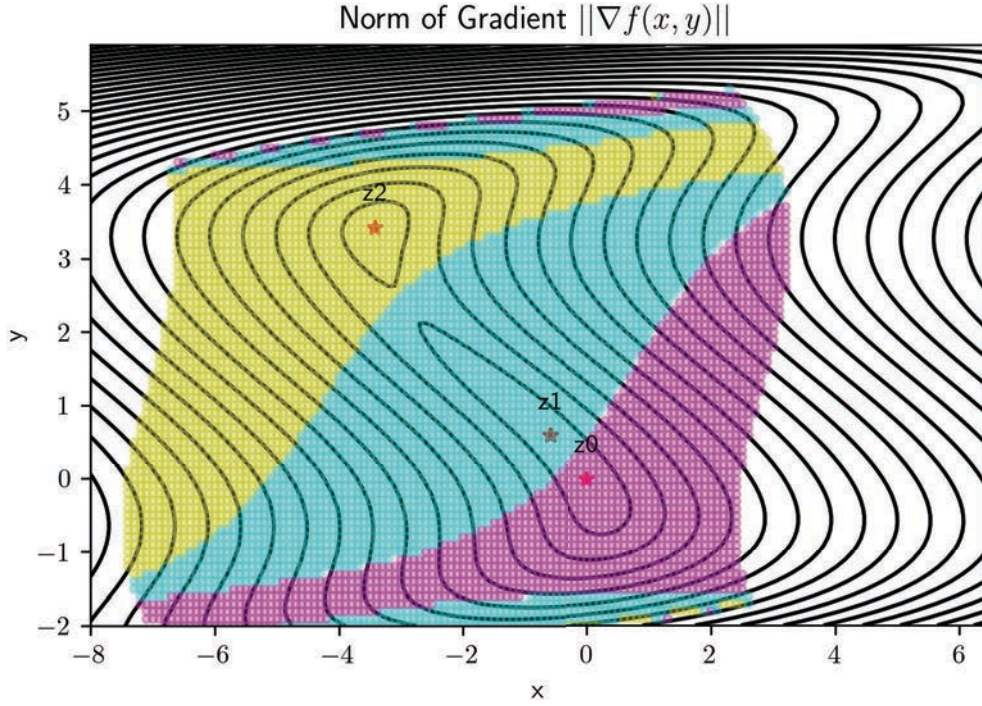


Figure 1: Contour graph $\|\Phi(x, y)\|$ (blue indicates the pool of attraction to the point z_1 , yellow to z_2 , purple to z_0)

2. The optimization problem for typical test functions, one of which is the Rosenbrock function $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$, which has a global minimum at the point of $(x, y) = (1, 1)$ is solved.

A comparative analysis of the results of the search for the extreme of the Rosenbrock function, which was provided as a neural network, researched and the most

popular methods for deep neural networks training (StoGD (Stochastic Gradient Decent) algorithm with momentum? and the Nesterov method, the Adam algorithm), using Facebook's PyTorch library, which contains all gradient optimization algorithms was performed. Numerous experiments showed the advantage of the developed method (New SGD algorithm with control) both in the rate of convergence and in the smoother behavior of the loss function (Fig. 2).

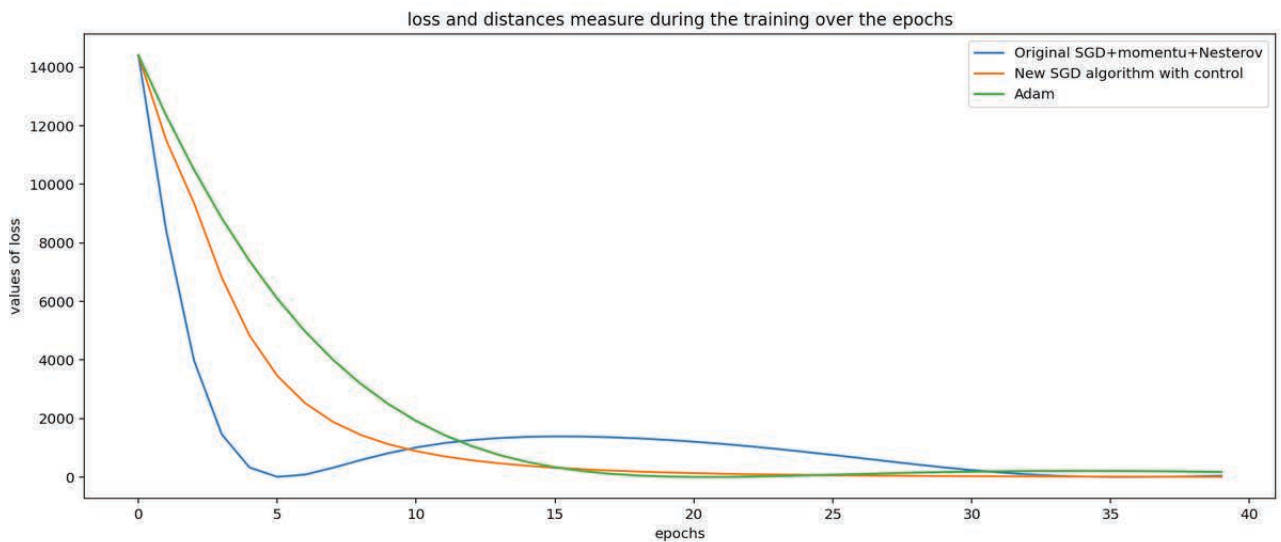


Figure 2: Graph of loss function when searching for an extreme function

3. Training multilayer perceptron for handwriting recognition.

MNIST (60,000 manually marked images) was used as a training and test dataset. Analysis of the multilayer perceptron learning error function in handwriting recognition showed that the developed method reduces the training time (up to three times) at a lower error rate (Fig. 3).

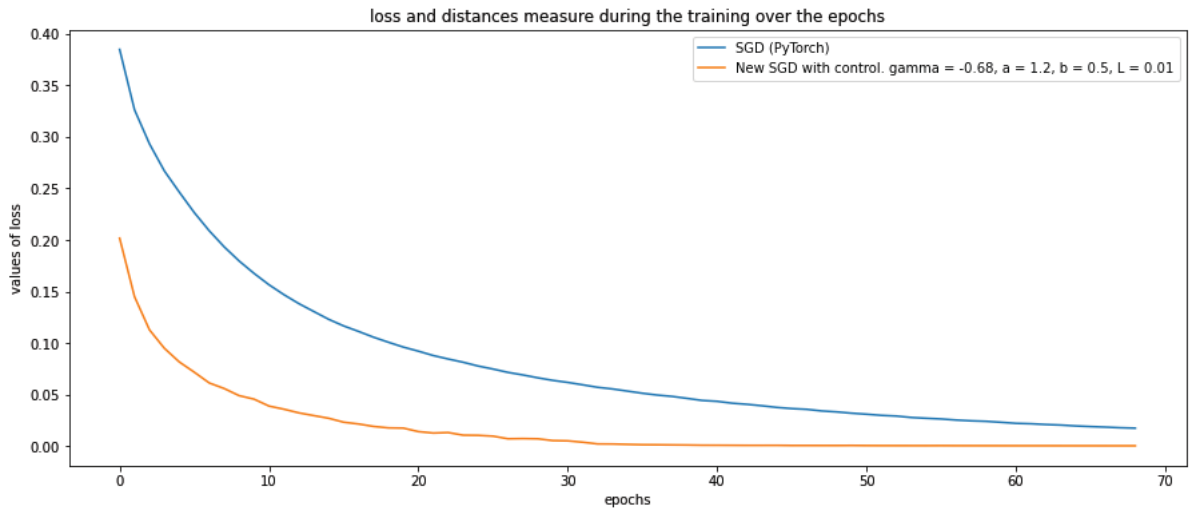


Figure 3: Graph of handwriting recognition learning error function

Thus, the methods of neural networks training were further developed based on the proposed modifications of the gradient descent method, which reduced the training time of neural networks of different architectures in solving problems of segmentation and classification of objects (**the third point of novelty of the scientific research**).

In **the fourth section** the software realization of the offered methods of neural networks training on the basis of nonlinear dynamics was developed and their approbation was carried out to build the subsystem of segmentation and classification of orthopantomograms for the dental intelligent information system.

When building the subsystem, the problem of converting *orthopantomograms* into odontograms, which are a specialized form of presenting information to a doctor about the condition of each patient's tooth, using a convolutional neural network, was solved.

At the first stage the training sample was formed. Since obtaining the necessary training sample is impossible due to the high cost of acquiring panoramic images, the need to clean them of extraneous noise and the difficulty of the initial manual segmentation by dentists, a method of training sample augmentation was developed using both

geometric and pixel transformations. This allowed creation of 300,000 images based on the initial set of 53 images and 32 binary masks. All images were saved as a multi-dimensional PyTorch tensor in a special internal format.

To solve the segmentation problem the architecture of the hybrid convolutional neural network was chosen based on the previously trained ResNet-18 network encoder and the decoder from the U-Net network. Training of this network involves the configuration of about 18 million parameters. As an error function, it is proposed to use a linear combination of cross-entropy and the Dice coefficient, which proved itself well in the training of convolutional neural networks. The training was conducted using the PyTorch library with various modifications of the gradient descent method. The use of GPU RAM, the minimum value of the loss function (Minimum) and the value of the loss function on the test dataset (Test) were studied (Table 1).

Table 1: Comparative analysis

Name	GPU memory (MiB)	Minimum	Test
RMSProps	3503	1.3766	0.5385
Adagrad	3571	0.0502	0.5269
Adadellta	3459	0.5520	0.5560
Adam	3459	0.0474	1.189
SGD	3450	0.2197	0.5365
New Method (3)	3471	0.2673	0.2049

The analysis showed that the proposed method, according to the learning error, has similar results in the training sample, and there is a decrease in the test. In addition, at the same cost of GPU RAM for the developed methods, the loss function is twice less on the test data compared to SGD and almost 7 times less compared to the Adam algorithm.

The graphs of the loss function during the neural network training (Fig 4) showed that the new method (3) minimizes the loss function relative to the SGD almost twice

as fast.

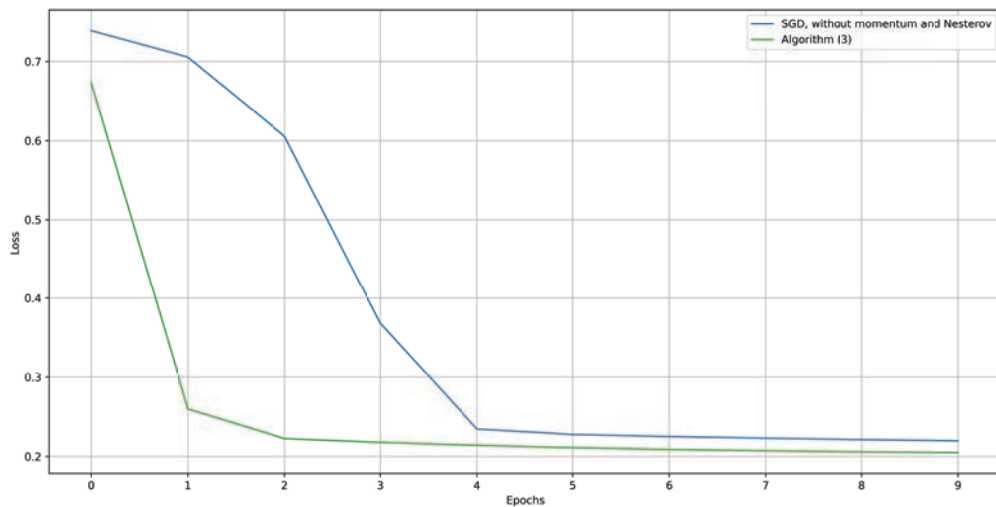


Figure 4: Graph comparing the minimization rate of the loss function of SGD algorithms (blue) and the New Method (3) (green) from the time of training

The conclusions formulate the scientific and practical results obtained during the work on the dissertation.

Keywords: methods of neural networks training, optimization, convolutional neural networks, segmentation, classification.

LIST OF THE APPLICANT'S PUBLICATIONS

The main materials of the dissertation and the main results of the research were published in 7 scientific papers (4 articles were published in the scientific and metric databases Scopus, three in the first quartile Q1; three articles were published in professional collections, 3 without co-authors) and in eleven approbation materials.

Scientific works in which the main scientific results of the dissertation are published

1. Dmitrishin D., Smorodin A., Stokolos A. On a family of extremal polynomials. *Comptes Rendus Mathematique*. 2019. 357(7). P. 591-596. DOI:<https://doi.org/10.1016/j.crma.2019.06.010>. (Scopus, Q1).

2. Dmitrishin D., Smorodin A., Stokolos A., Tohaneanu M. Symmetrization of Suf-fridge polynomials and approximation of T-symmetric Koebe functions. *Journal of Mathematical Analysis and Applications*. 2021. 503(3). DOI:<https://doi.org/10.1016/j.jmaa.2021.125350>. (Scopus, Q1).
3. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. *Applied and Computational Harmonic Analysis*. 2021. 56. P. 283-305. DOI:<https://doi.org/10.1016/j.acha.2021.08.008>. (Scopus, Q1)
4. Smorodin A. Predictive control methods in tasks of searching saddle points. *Proceedings of Odessa Polytechnic University*. 2020. V. 3(62). P. 80-90. DOI:[10.15276/opu.3.62.2020.10](https://doi.org/10.15276/opu.3.62.2020.10). URL:<https://pratsi.op.edu.ua/articles/show/20968>
5. Smorodin A. The use of control theory methods in training neural networks on the example of teeth recognition on panoramic x-ray images//*Automation of Technological and Business Processes*. 2021. V. 13(2). P. 36-40. URL:<http://dx.doi.org/10.15673/atbp.v13i2.2055>
6. Smorodin, A.V. Using control theory methods in the neural networks study on the example of handwritten text. *Information technology applied aspects*. V. 4(3). 2021. P. 243–249. URL:<http://aait.ccs.od.ua/index.php/journal/article/view/119>
7. Dillies, J., Dmitrishin, D., Smorodin, A., Stokolos A. (2022). On the Koebe Quarter Theorem for Polynomials. *Proceedings of the International Geometry Center*, V. 14(3). 2022. P. 219-230. <https://doi.org/10.15673/tmgc.v14i3.2057>. (Scopus, Q4)

Scientific works that certify the approbation of the dissertation materials

8. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. Harmonic Analysis and Partial Differential Equations. Canadian Mathematical Society. 2021.
9. Dmitrishin D., Smorodin A., Stokolos A. Extremal polynomials and geometric complex analysis. Algebraic and Geometric Methods of Analysis. International Online Conference. 2020.
10. Smorodin A. On question by Dimitrov. Analysis Seminar. Odessa. I. I. Mechnikov National University. 2018.
11. Smorodin A. Predictive control methods to solve problems of finding saddle points and neural networks training. Current issues of physical, mathematical and engineering sciences: theoretical and applied research. 2021. P. 13-23. II International scientific and practical Internet conference. URL: https://open.scilab.org/wp-content/uploads/2021/05/aktualni-pitannja-fiziko-matematichnih-ta-tehnichnih-nauk-teoretichni-ta-prikladni-doslidzhennja_2021_05_21.pdf
12. Smorodin A. Forecontrol methods at optimization problems solving. Theoretical and practical problems of using mathematical methods and computer-based technologies in education and science. 2021. III All-Ukrainian scientific-practical online conference.
13. Dmitrishin D., Stokolos A., Smorodin A. On a new family of extremal polynomials. The 35th annual Southeastern Analysis. Meeting, University of Alabama, 2019.

14. Dmitrishin D., Stokolos A., Smorodin A. An extremal problem for polynomials. CMS 75th +1 Anniversary Summer Meeting, Montreal, 2021
15. Dmitrishin D., Skrinnik I., Smorodin A., Stokolos A. Dimitrov's question for the polynomials of degree 1,2,3,4,5,6. 2018. arXiv:1808.08636. URL: <https://arxiv.org/abs/1808.08636>.
16. Dmitrishin D., Smorodin A., Stokolos A. Estimating the Koebe radius for polynomials. 2018. arXiv: 1805.06927. URL: <https://arxiv.org/abs/1805.06927v1>
17. Dmitrishin D., Smorodin A., Stokolos. On C. Michel's hypothesis about the modulus of typically real polynomials. 2020. arXiv: 2005.12432. URL: <https://arxiv.org/abs/2005.12432>
18. Dillies J., Dmitrishin D., Smorodin A., Stokolos. On the Koebe Quarter Theorem for Polynomials. 2019. arXiv: 1904.11039. URL: <https://arxiv.org/abs/1904.11039>

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	32
ВСТУП	33
РОЗДІЛ 1 АНАЛІЗ ЗАДАЧ СЕГМЕНТАЦІЇ ТА КЛАСИФІКАЦІЇ ОБ’ЄКТІВ НА ЗОБРАЖЕННЯ ІЗ ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	39
1.1 Особливості застосування нейронних мереж при сегментації та класифікації об’єктів	39
1.2 Аналіз методів навчання глибоких нейронних мереж	48
1.3 Подання методу градієнтного спуску як дискретної динамічної системи	55
1.4 Висновки до першого розділу	64
РОЗДІЛ 2 МЕТОД ПОШУКУ ПАРАМЕТРІВ ДЛЯ ЗАПРОПОНОВАНИХ МОДИФІКАЦІЙ МЕТОДУ ГРАДІЄНТНОГО СПУСКУ	65
2.1 Зв’язок задачі пошуку та стабілізації періодичних точок із задачами геометричного комплексного аналізу	65
2.2 Оптимальні многочлени та їх властивості	68
2.3 Предикативний контроль	105
2.4 Модифікаційні схеми градієнтного спуску	110
2.5 Висновки по другому розділу	113
РОЗДІЛ 3 МЕТОДИ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ МОДИФІКАЦІЙ МЕТОДУ ГРАДІЄНТНОГО СПУСКУ	114
3.1 Пошук екстремальних та сідлових точок	114

3.2	Тестові функції	116
3.3	Навчання багат шарового персептрона для розпізнавання рукописних цифр	119
3.4	Висновки до третього розділу	123
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНИХ МЕТОДІВ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ НЕЛІНІЙНОЇ ДИНАМІКИ		127
4.1	Перетворення ортопантомограм в одонтограми	127
4.2	Підготовка даних для навчання нейронної мережі	133
4.3	Вибір архітектури нейронної мережі	137
4.4	Навчання нейронної мережі	139
4.5	Висновки до четвертого розділу	143
ЗАГАЛЬНІ ВИСНОВКИ		145
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		147
ДОДАТОК А Відомості про апробацію результатів дисертації		160
ДОДАТОК Б. Довідка про використання результатів дисертаційної роботи у науково-дослідницької діяльності		162
ДОДАТОК В. Довідка про впровадження результатів дисертаційної роботи у навчальний процес		163
ДОДАТОК Г. Список публікацій здобувача за темою дисертації		164

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГС – Градієнтний Спуск

ДДС – Дискретна Динамічна Система

ІАД – Інтелектуальний Аналіз Даних

ММН – Методи Машинного Навчання

AI – Artificial Intelligence

CNN – Convolutional Neural Network

CPU – Central Processing Unit

GPU – Graphics Processing Unit

JPG – Joint Photographic Experts Group

SGD – Stochastic Gradient Descent

SVN – Support Vector Machine

TPU – Tensor Process Unit

VGG – Visual Geometry Group

ВСТУП

Актуальність теми. У теперішній час для вирішення завдань сегментації та класифікації об'єктів на зображеннях, які затребувані в інтелектуальних інформаційних системах для різних сфер людського життя (наприклад, у медицині, маркетингу, логістиці та інш.), все частіше використовуються глибокі нейронні мережі (НМ), які інтегровані всередину архітектури таких систем. За даними компанії Bloomberg за період з 2012 по 2015 роки, тільки на хмарній платформі компанії Google кількість проектів, що використовують нейронні мережі, збільшилася в три тисячі разів. Однак, незважаючи на ефективність проектів, побудованих на глибоких нейронних мережах складної архітектури, їх розробники та дослідники зіткнулися із закономірною проблемою – збільшенням кількості обчислювальних ресурсів, що витрачаються на їхнє навчання (за даними компанії OpenAI спостерігається трьохсот тисячне збільшення витрат на навчання).

Проведений аналіз показав, що одним із ефективних шляхів вирішення зазначеної проблеми є зменшення часових витрат на навчання нейронних мереж, що може бути досягнуто за рахунок пошуку нових та удосконалення існуючих методів оптимізації, які застосовуються при навчанні НМ. Найбільш поширенішими при навчанні глибоких НМ є низка методів, в основі яких лежить градієнтний метод оптимізації. Однак усім їм властивий в умовах багатоцільової, мульти-модальної та зашумленої функції загальний недолік – «повільна» збіжність, яка суттєво збільшує час навчання. Для його усунення запропоновано модифікацію методів навчання НМ на основі градієнтного спуску, що ґрунтується на застосуванні теорії нелінійної динаміки, та розроблено відповідні теоретичні положення для реалізації методів.

Таким чином, дисертаційна робота, що присвячена розробці методів навча-

ння нейронних мереж на основі положень нелінійної динаміки, є актуальною та своєчасною.

Зв'язок роботи з науковими програмами, планами, темами. Дисертацію виконано згідно тематичних планів НДР ОНПУ за період 2017 - 2021 років під час виконання бюджетної теми № 716-145 – «Інноваційна інтелектуальна інформаційна технологія аналізу та синтезу енергоефективних гідроаеродинамічних елементів та систем» (№ ДР 0121О111720).

Мета і задачі дослідження Метою роботи є скорочення часу навчання нейронних мереж під час вирішення завдань сегментації та класифікації.

Для досягнення мети вирішено такі завдання:

- 1) проаналізовано проблеми скорочення часу під час навчання нейронних мереж під час вирішення завдань сегментації та класифікації;
- 2) удосконалено методи градієнтного спуску на основі застосування положень нелінійної динаміки;
- 3) розроблено схеми реалізації вдосконаленого методу градієнтного спуску;
- 4) удосконалено методи навчання нейронних мереж для різних архітектури НМ;
- 5) розроблено програмні засоби, які реалізують запропоновані методи, та проведена їх апробація на практичних завданнях.

Об'єктом дослідження є процес навчання нейронних мереж.

Предметом дослідження є методи навчання нейронних мереж на основі нелінійної динаміки.

Методи дослідження. У процесі виконання дисертаційного дослідження використано: методи теорії функцій комплексного змінного та геометричного ком-

плексного аналізу, методи теорії управління нелінійними дискретними динамічними системами, методи теорії оптимізації та системного аналізу, методи глибокого машинного навчання, методи інтелектуального аналізу даних.

Наукова новизна одержаних результатів полягає в удосконаленні методів навчання глибоких нейронних мереж, а саме:

- *удосконалено* метод градієнтного спуску на основі положень нелінійної динаміки, що дозволило розробити модифікації методу з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС та застосувати його як при навчанні нейронних мереж з метою зменшення часу навчання так і для вирішення інших оптимізаційних задач;
- *вперше* розроблено метод пошуку параметрів для запропонованих модифікацій методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС на основі розв'язання оптимізаційних задач на класі комплексних поліномів, який дозволяє забезпечити їх реалізацію для широкого кола оптимізаційних задач;
- *набули подальшого розвитку* методи навчання нейронних мереж на основі запропонованих модифікацій методу градієнтного спуску, що дозволило зменшити час навчання нейронних мереж різних архітектур при вирішенні задач сегментації та класифікації об'єктів.

Практичне значення одержаних результатів полягає в наступному:

- розроблено алгоритми реалізації удосконалених методів градієнтного спуску на основі положень нелінійної динаміки;
- вирішено низку оптимізаційних завдань на класі комплексних поліномів, що дозволило забезпечити практичну реалізацію удосконалених методів градієнтного спуску на основі положень нелінійної динаміки;

- розроблено програмні модулі, які розширили оптимізаційний блок бібліотеки PyTorch за рахунок удосконалених методів градієнтного спуску на основі положень нелінійної динаміки;
- спроектовано та впроваджено підсистему сегментації панорамних рентгенівських знімків з використанням запропонованих рішень як складову стоматологічної інформаційної системи;
- прискорено час навчання нейронної мережі для вирішення завдань сегментації та класифікації зубів на панорамних рентгенівських знімках відносно SGD алгоритму.

Основні результати дисертаційного дослідження використано для навчання глибокої нейронної мережі при сегментації зубів на панорамних рентгенівських знімках та впроваджено в компанії Artifakt AI (Франція, Нант), що підтверджено відповідним актам впровадження (Додаток А). Отримані результати впроваджено у навчальний процес кафедри прикладної математики та інформаційних технологій Державного університету «Одеська політехніка» (довідка про впровадження від 04012022) та у науково-дослідну роботу Державного університету «Одеська політехніка» (довідка про впровадження від 04012022).

Персональний внесок здобувача. Результати, які складають основний зміст дисертаційної роботи отримані автором самостійно.

У наукових статтях, опублікованих у співавторстві, автору належать:

[1] – доведення леми Д відносно екстремального значення множини поліномів, лема Е (має технічний характер), та експериментальна перевірка властивостей сімейства поліномів в теоремі 1.

[2] – теорема щодо апроксимації симетричної функції Кебе та всі численні експерименти.

[3] - ілюстрація загального результату на прикладі окремого випадку поліномів, також лема 1, лема 2 та лема 5, які носять технічний характер.

[7] - перевірка сформульованої гіпотези для поліномів малих ступеней, тобто пункт 4 статті.

Апробація результатів дисертації. Основні положення роботи та окремі її розділи доповідалися і отримали схвалення на науково-технічних конференціях професорсько-викладацького складу Одеського національного політехнічного університету 2017-2020 рр.

Також результати дисертації отримали апробацію на наступних конференціях:

Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. 2021, Harmonic Analysis and Partial Differential Equations. Canadian Mathematical Society.

Dmitrishin D., Smorodin A., Stokolos A. Extremal polynomials and geometric complex analysis. 2020. Algebraic and Geometric Methods of Analysis. International Online Conference.

Smorodin A. On question by Dimitrov. 2018. Analysis Seminar. Odessa. I. I. Mechnikov National University.

Смородин А. Методи прогнозного управління в задачах пошуку сидлових точок та навчання нейронних мереж. Актуальні питання фізико-математичних та технічних наук: теоретичні та прикладні дослідження. 2021. II Міжнародна науково-практична інтернет-конференція.

Смородін А. Методи прогнозного управління для задач оптимізації. Теоретико-практичні проблеми використання математичних методів та комп'ютерно-орієнтованих технологій в освіті та науці. 2021. III Всеукраїнська науково-практична

онлайн-конференція.

Dmitrishin D., Stokolos. A, Smorodin A. On a new family of extremal polynomials. The 35th annual Southeastern Analysis. Meeting, Georgia Southern University, 2019.

Dmitrishin D., Stokolos. A, Smorodin A. An extremal problem for polynomials. CMS 75th +1 Anniversary Summer Meeting. Georgia Southern University, 2021

Основні результати дисертаційної роботи викладено в 7 публікаціях, з них: 3 статей у наукових фахових виданнях України з технічних наук, 4 статті у наукових періодичних виданнях з напрямку, з якого підготовлено дисертацію, що індексується наукометричною базою SCOPUS, 7 публікацій у працях і матеріалах наукових конференцій, 4 статті у формі препринту.

Структура і обсяг роботи. Дисертація має загальний обсяг 166 сторінок і складається з вступу, чотирьох розділів, висновків, списку використаних джерел і 4 додатків. Основний текст дисертації викладено на 113 сторінках. Робота містить 32 рисунків і 4 таблиць. Список використаних джерел становить 99 найменувань.

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧ СЕГМЕНТАЦІЇ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯ ІЗ ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

1.1 Особливості застосування нейронних мереж при сегментації та класифікації об'єктів

Разом з розвитком систем зберігання та збору даних, а також технологій Big Data виникла потреба у швидкій обробці цих даних та отримання з них нової інформації, необхідної для прийняття рішень та пошуку прихованих закономірностей за рахунок створення сучасних математичних моделей та методів. Серед різних типів даних особливе місце займають зображення, що отримані за допомогою сучасних мобільних пристроїв, супутників, медичних апаратів. Базовими задачами обробки зображень є сегментація та класифікація. При сегментації та класифікації зображень для побудови математичних моделей на основі емпіричної інформації (за даними одержаних зображень об'єктів, часових рядів, мовних послідовностей, наборів сенсорних даних різної фізичної природи) найбільш розповсюдженим підходом є застосування штучних нейронних мереж, що в випадку великих обсягів даних потребує удосконалення методів машинного навчання.

На даний час методи машинного навчання (ММН) досягли значного рівня. ММН здатні вирішувати широке коло завдань в області інтелектуального аналізу даних – сегментації, класифікації, прогнозування тощо. Характерною рисою ММН є не пряме розв'язання задачі, а навчання в процесі врахування множини подібних задач. Навчання зводиться до оптимізації (мінімізації) функцій втрат. Основним методом оптимізації функцій, що застосовуються у задачах машинного навчання, є метод градієнтного спуску (ГС) [61, 83].

Головними недоліками цього методу при навчанні нейронних мереж є, по

перше, різке падіння швидкості навчання на ділянках незначної зміни функції помилок («плато»), при цьому теоретично ГС завжди знаходить локальний мінімум функції. Другим суттєвим недоліком, який є наслідком першого, є значний час пошуку, що особливо відчувається на великих даних [84]. Таким чином, одним з найважливіших завдань для розвитку штучних нейронних мереж є удосконалення ММН, яке вирішується у даній роботі.

Розглянемо типові архітектури нейронних мереж, які для багатьох завдань сегментації та класифікації зображень змогли показати точність достатню для низки прикладних завдань [55, 59, 65, 81, 82, 88].

Першою архітектурою нейронних мереж, яка могла розпізнавати рукописні літери, був багат шаровий перцептрон, розроблений Френком Розенблаттом у 1958 році. Дана мережа складається з вхідного та вихідного шарів та кількох прихованих та навчальних шарів пов'язаних між собою нейронів. Перцептрон це нейронна мережа, що має такі властивості [82]:

- один або кілька прихованих шарів;
- визначена порогова функція;
- сигнал розповсюджується у прямому напрямку від вхідного шару до вихідного.

У третьому розділі вирішена задача розпізнавання рукописних цифр з використанням багат шарового перцептрону та запропонованого методу навчання.

За останні роки велику популярність набули сегментація та класифікація зображень за допомогою згорткових нейронних мереж [93]. Особливістю цього підходу є віднесення (анотування) кожного пікселя до того чи іншого класу об'єктів. Процес анотування зображень вимагає значних обчислювальних витрат, він спрощує завдання сегментації, проставляючи цільову ознаку (ground truth).

Цією ознакою може бути представлення зображення пікселями, що отримали репрезентативний колір певного об'єкта. Кожен клас об'єктів зазвичай представляється репрезентативним кольором, щоб цільовий об'єкт можна було відобразити на сегментованому зображенні.

Глибоке навчання – це розділ машинного навчання, який заснований на наборі методів, які моделюють багаторівневі абстракції даних, використовуючи глибокий граф з кількома нейронами, що виконують лінійні або нелінійні перетворення [49, 68].

Найкращі результати в галузі класифікації об'єктів у зображеннях показала згорткова нейронна мережа – Convolutional Neural Network (CNN). Результат класифікації об'єктів на основі CNN, на відміну від багаторівневого персептрона, покращується за рахунок виділення двомірної топології зображення.

Згорткові нейронні мережі забезпечують часткову стійкість щодо змін масштабу, поворотів, зміни фокусу та інших геометричних перетворень.

Починаючи з 2012 року, згорткові мережі займають перші місця у міжнародному конкурсі з розпізнавання образів ImageNet [52]. Цей тест спрямований на порівняння властивостей різних архітектур нейронних мереж на задачах розпізнавання та класифікації зображень. Серед переможців різних років можна назвати мережі: AlexNet, VGG-16, GoogleLeNet та ResNet. Розглянемо особливості застосування та архітектури цих мереж.

AlexNet [59] була першою глибокою CNN мережею, яка перемогла у змаганні ILSVRC-2012 з точністю 84.6 відсотків у TOP-5 тесті порівняно з 73.8 відсотками свого найближчого конкурента, який використав стандартні методи сегментації (рис. 1.1).

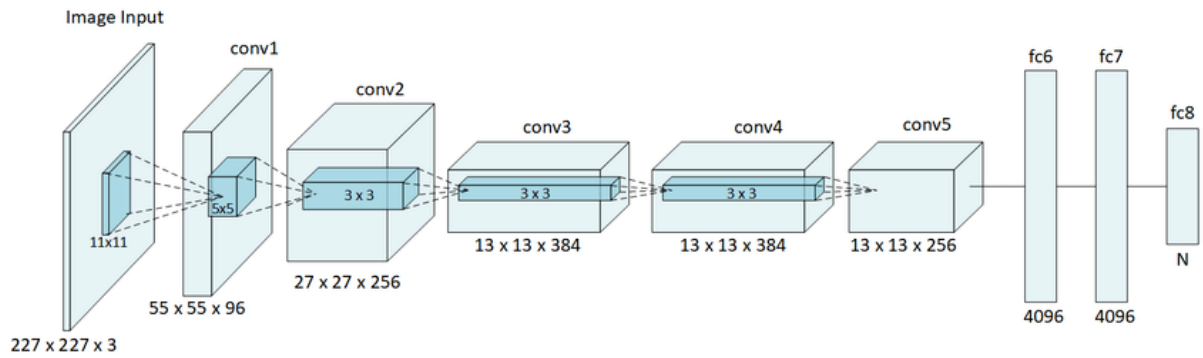


Рисунок 1.1: Спрощена архітектура мережі AlexNet

Наступне покращення архітектури мережі під назвою VGG (Visual Geometry Group) [88], яке створене Visual Geometry Group університету Оксфорд. Її 16-рівнева версія VGG-16 перемогла у змаганні ILSVRC-2013 з результатом 92.7 відсотка (рис. 1.2).

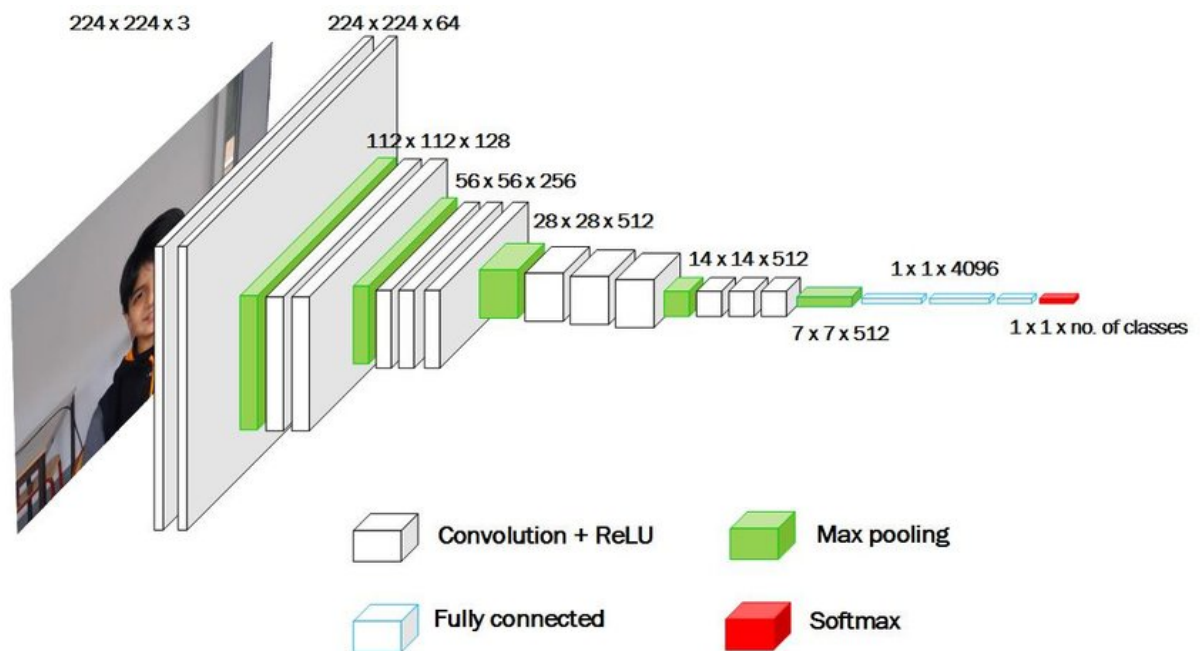


Рисунок 1.2: Архітектура мережі VGG-16

У 2014 році переможцем виявилася мережа GoogLeNet, яка запропонована Szegedy et al., котра забезпечила точність розпізнавання 92.3 відсотка. Її глибина досягала 22 шарів. Нова частина, що названа "inception" модулем, додала скла-

дності у конструкцію мережі, але дозволила з'єднувати різні CNN шари різними шляхами, а не лише послідовно (рис. 1.3).

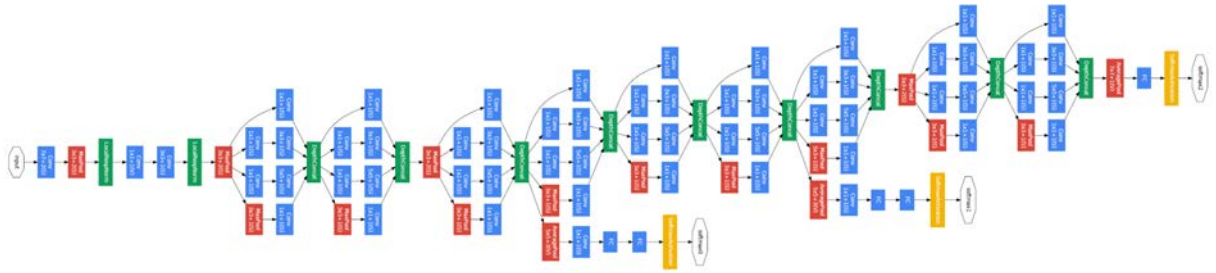


Рисунок 1.3: Архітектура мережі GoogLeNet

Через два роки в 2016 році Microsoft розробила нову архітектуру ResNet [55], яка має кілька модифікацій з різною кількістю шарів. Головний модуль цієї мережі дає змогу пропустити кілька з'єднань усередині мережі. Це дозволяє уникнути нульових значень градієнтів та уникнути погіршення точності при додаванні нових шарів в особливо глибоких нейронних мережах (рис. 1.4).

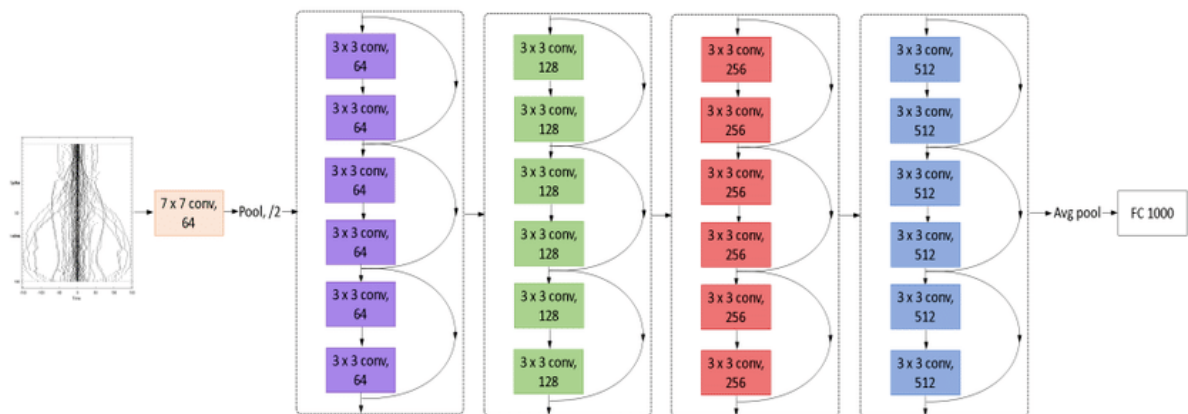


Рисунок 1.4: Архітектура мережі ResNet-50

Всі зазначені приклади показують, що кожне з покращень результатів вимагає все більш глибоких мереж, які складно навчити на невеликих обсягах тренувальних даних, доступних у синтетичних тестах типу ILSVRC-2012, але недоступних у багатьох реальних завданнях.

Розглянуті архітектури є основою розв'язання задач сегментації та класифікації об'єктів, де найкращу продуктивність показують сучасні нейронні мережі (англ. fully-convolutional networks) [65] (рис. 1.5).

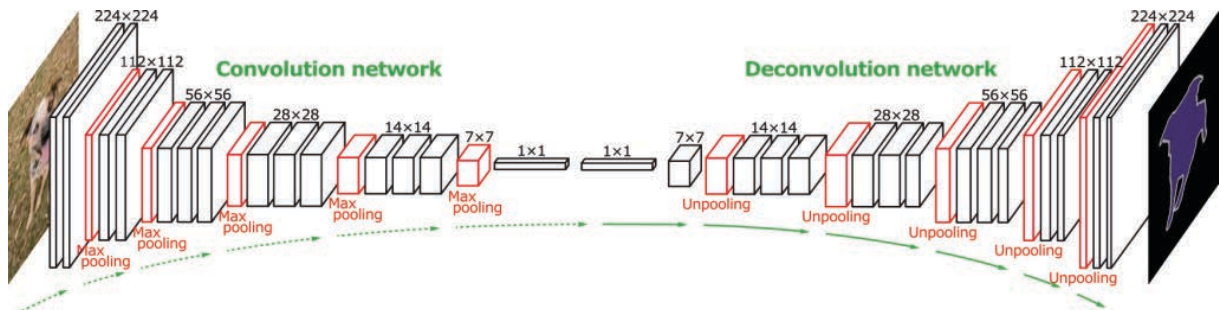


Рисунок 1.5: Архітектура «fully-convolutional networks» мережі

Архітектура цих мереж складається з двох частин: конволюційної мережі та деконволюційної мережі. Перша з них заснована на одній із архітектур, що використовуються в задачах класифікації зображень. При цьому конволюційну частину часто ініціалізують з перетренованими вагами, щоб оптимізувати час навчання та використовувати знання, набуті під час навчання на великих обсягах зображень високої якості.

У медицині часто неможливо зібрати достатню кількість знімків та підготувати високоякісний набір сертифікованих лікарями даних. Це вимагало створення спеціальної архітектури, яка потребує меншого обсягу даних для завдання семантичної сегментації - U-Net [41, 81], що розроблена в 2015 році Olaf Ronneberger, Philip Fischer та Thomas Brox в університеті Фрайбург-ім-Брайсгау, Німеччина (рис. 1.6).

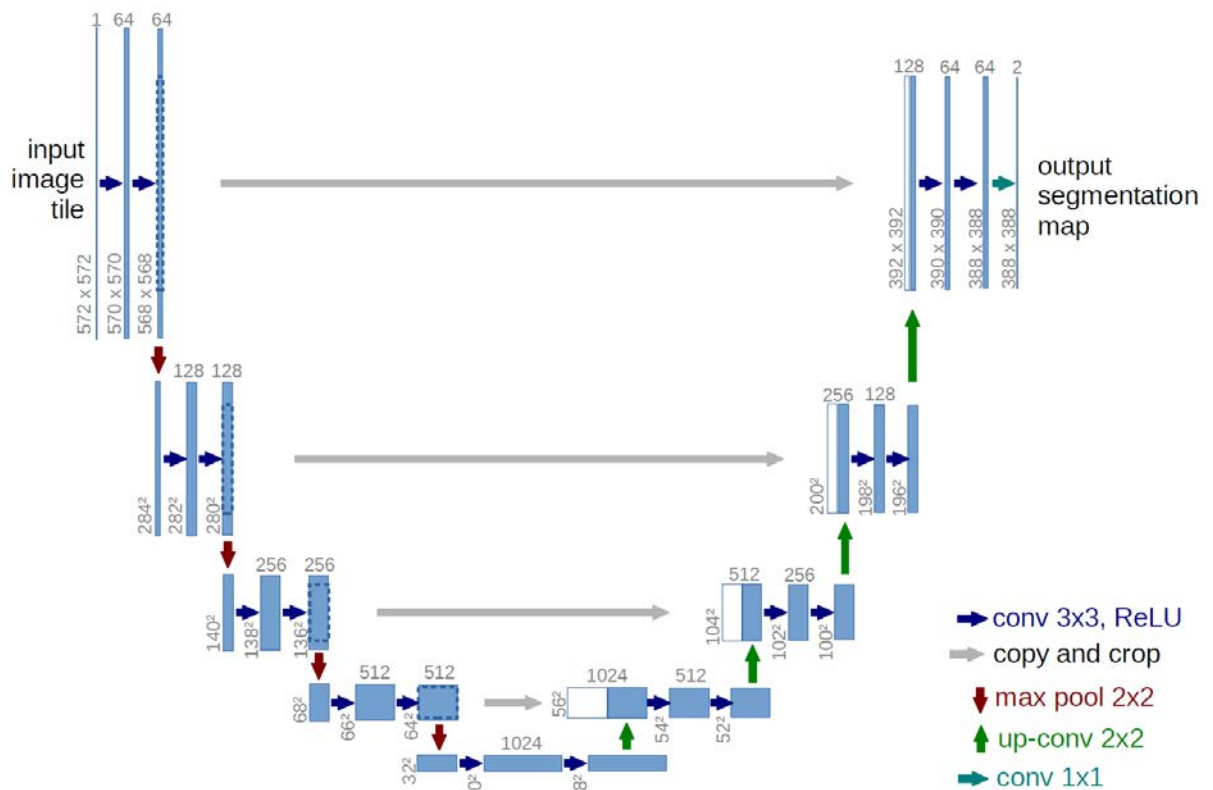


Рисунок 1.6: Архітектура мережі U-Net

Назва мережі вийшла з її U образної архітектури, яка містить дві різні частини: Даунсамплінг та Апсамплінг, з'єднаних між собою для передачі інформації. Даунсамплінг переважно виділяє особливості із зображень, коли Апсамплінг – локалізує об'єкти. Завдяки невеликій глибині мережі та передачі додаткової інформації між шарами на практиці вдається провести навчання цієї мережі на великому обсязі даних.

Крім завдання сегментації об'єктів, часто потрібно вирішувати завдання визначення окремих екземплярів об'єктів. Для цих цілей була розроблена мережа R-CNN (регіональна конволюційна мережа) [42], яка вирішує одночасно завдання локалізації та семантичної сегментації об'єктів. Її розширення Faster R-CNN використовує regional proposal network визначення області інтересу (Region of Interest) та знаходження прямокутника, який містить необхідний об'єкт у собі (рис. 1.7).

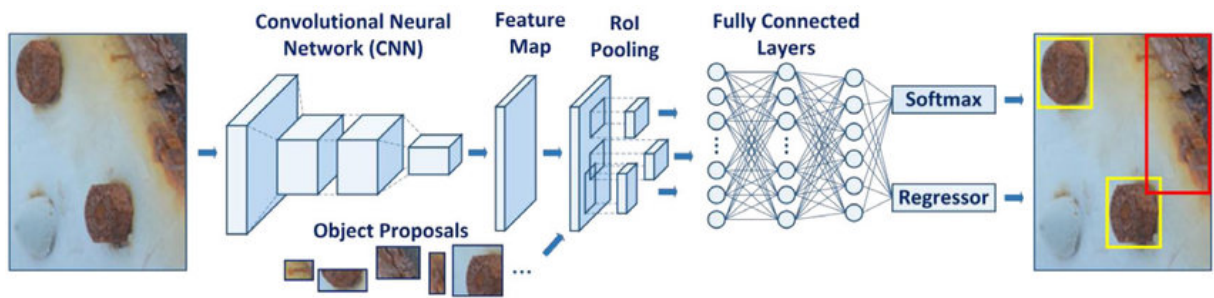


Рисунок 1.7: Архітектура мережі Fast-R-CNN

Дослідження компанії Grandviewresearch щодо зростання інтересу та використання їхніх ресурсів у проектах глибокого навчання з 2017-2028 років показали бурхливе зростання цього розділу штучного інтелекту (рис. 1.8).

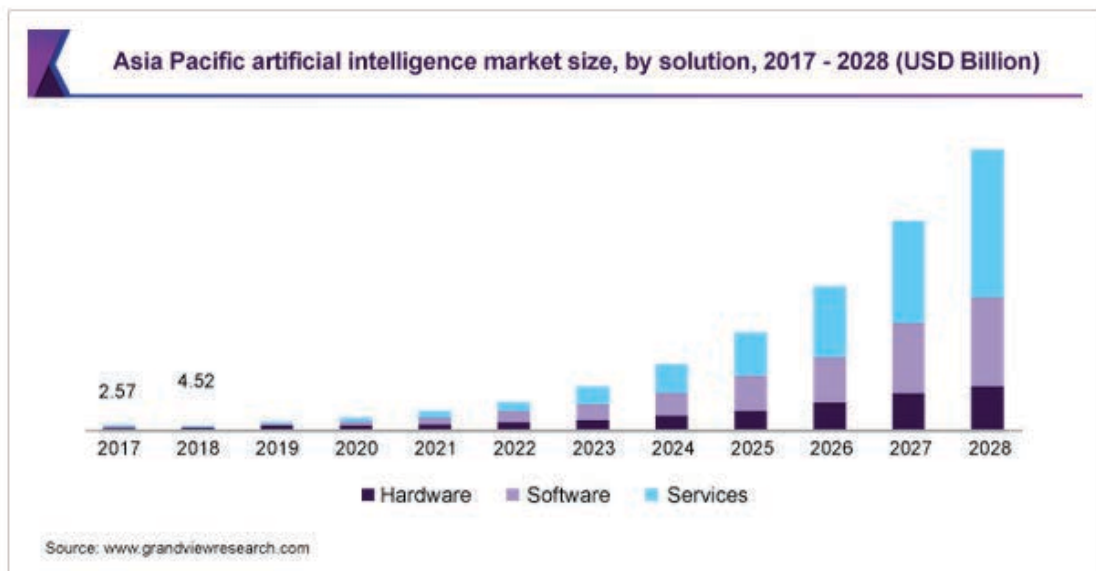


Рисунок 1.8: Звіт Grandviewresearch за 2017-2028 роки з Artificial Intelligence розміру ринку

З 2015 року зростання тільки прискорювалося з появою інтересу, що збільшується з боку представників різних індустрій, включаючи медичну, розважальну, технологічну та виробничу.

Разом із зростанням інтересу збільшилися вимоги до ресурсів, необхідних для цих досліджень і створення різних продуктів з урахуванням нейронних мереж. Останні дослідження [54, 84] показали 300 тис. кратне зростання ресурсів з 2012

по 2018 роки (рис. 1.9).

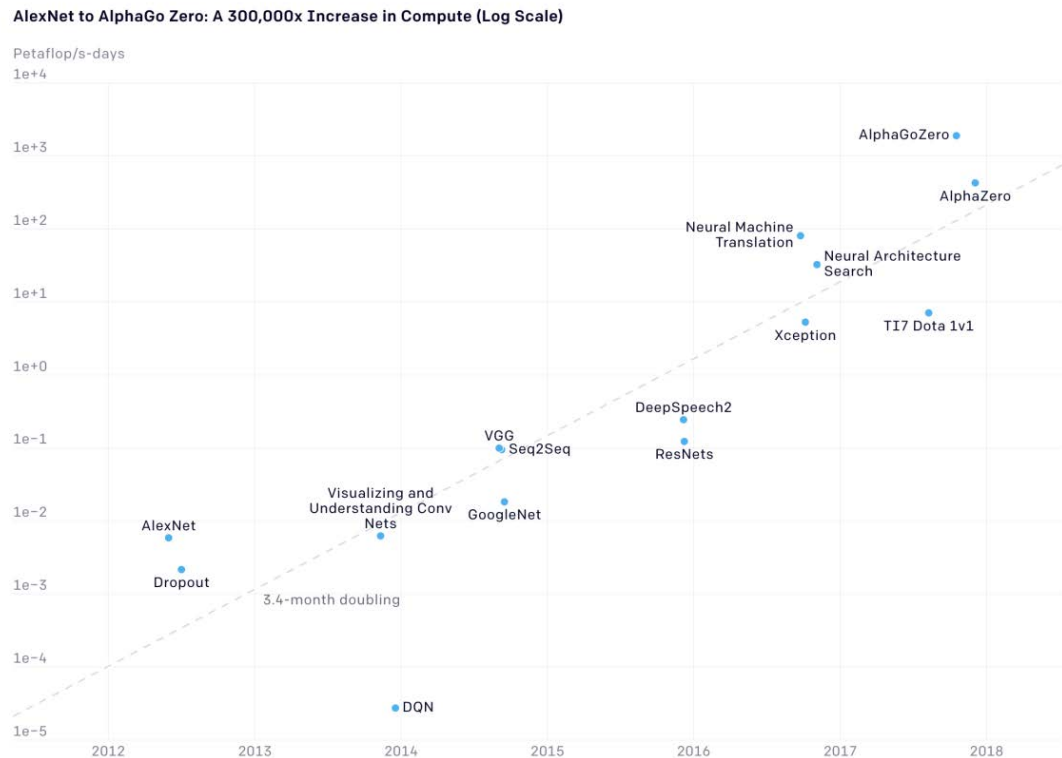


Рисунок 1.9: Діаграма залежності кількості обчислень від різних архітектур мереж в процесі навчання

Таке зростання кількості обчислень робить актуальним вирішення завдання підвищення швидкодії процесу навчання нейронних мереж, іноді навіть попереду точності самої моделі, оскільки навчання однієї нейронної мережі до рівня сумісного з рівнем спеціаліста у певній предметній області може коштувати близько 10 мільйонів доларів США, витрачених на електроенергію (сюди не включається вартість обладнання та витрати на розробку нейронної мережі).

Прикладом може бути останній варіант мережі AlphaGo [87], навчений з урахуванням методу Монте-Карло, який застосував підрозділ Google (DeepMind) для навчання нейронних мереж гри в Go. Цю мережу тренували грою в Go саму з собою протягом 40 днів на 4 TPU (Tensor Process Unit, спеціальний процесор компанії Google, створений для тренування нейронних мереж). Ці приклади пока-

зують практичну важливість прискорення процесу навчання глибоких нейронних мереж, що важливо особливо на великих обсягах даних.

З цього випливає, що навіть невелике зменшення часу навчання нейронної мережі є актуальним, важливим, а іноді і першорядним для Artificial Intelligence (AI) проектів.

Як правило, створення AI проекту на основі ММН передбачає наступні основні кроки:

- збір та обробка тренувальної інформації;
- створення архітектури нейронної мережі під певне завдання (може бути доступна в бібліотеках);
- навчання мережі, що передбачає використання обраного метода оптимізації на тренувальних даних;
- перевірка результатів класифікації даних тестового набору, які не брали участь у процесі навчання.

Таким чином, в циклі навчання використовується обраний метод оптимізації, від якого залежить швидкість навчання нейронної мережі. Оптимізація полягає в знаходженні локального мінімуму складної та багатовимірної функції. Отже, швидкість процесу навчання напряму залежить від швидкодії оптимізації. Аналіз показав, що найбільш поширеними при навчанні нейронних мереж є методи ГС та їх модифікації.

Розглянемо такі методи, які використовуються в ММН.

1.2 Аналіз методів навчання глибоких нейронних мереж

Завданням процесу навчання нейронних мереж є підбір таких внутрішніх параметрів, щоб отримати найкраще наближення у передбаченні як на тренуваль-

ному, так і на тестовому наборі даних. Якість результату вимірюється функцією, яка вимірює помилку очікуваного результату від передбаченого. Отже, функція відхилення залежить як від одного з прикладів $\bar{x}_i$ та і від параметрів обраної нейронної мережі Θ .

У практиці використовують множину функцій втрат (цільових функцій), спеціально створених для різних класів завдань.

Завдання навчання нейронної мережі зводиться до пошуку мінімуму функції втрат. Отже, весь процес навчання можна уявити у вигляді стандартного оптимізаційного завдання, яке можна вирішувати відомими оптимізаційними методами. Множину цих методів можна розбити на два великі класи: першого та другого порядків. Відмінність між класами полягає у ступені похідної цільової функції, яку використовують для оптимізації.

Обчислення перших та других похідних призначаються для функції втрати, яка завдяки залежності від параметрів мережі є багатовимірною. Основною перевагою методів другого порядку є можливість враховувати локальну кривизну функції, яка дозволяє їм легше справлятися з сідловими точками та поверхнями типу «яр». Однак, так як кількість параметрів нейронів перебуває в інтервалі між 10 мільйонами та 1-2 мільярдами, то методи другого порядку практично не використовують при навчанні нейронних мереж через їхню додаткову обчислювальну ресурсоємність.

Розглянемо найпопулярніші методи першого порядку, що часто використовуються для вирішення задач лінійної регресії та класифікаційних задач. Як зазначалось вище, вони, як правило, засновані на методі градієнтного спуску. Для цього введемо деякі позначення. Нехай N буде потужністю множини навчальних даних. Сам цей набір має такий вигляд

$$\{(\bar{x}_i, \bar{y}_i)\}, \text{ де } i = 1, \dots, N.$$

Обидва елементи пари є векторами деяких просторів різних розмірностей. Зазвичай це багатовимірні дійсні простори.

Позначимо нейронну мережу функцією $f(\bar{x}; \Theta)$, а цільову функцію - $L(\bar{x}, \bar{y})$.

Завдання полягає у підборі таких параметрів нейронної мережі Θ , які мінімізують усереднене значення цільової функції $\frac{1}{N} \sum_i L(f(\bar{x}_i; \Theta), \bar{y}_i)$ по всьому навчальному набору даних.

Розглянемо базовий метод *градієнтного спуску*. Для цього на кожному кроці випадковим чином виберемо підмножину розміру B з прикладів, що навчаються $\{\bar{x}_1, \dots, \bar{x}_B\}$. Це число B називають розміром пакета. У цих позначеннях метод можна записати у вигляді

$$\Theta_{t+1} = \Theta_t - \frac{1}{B} \gamma \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta_t), \bar{y}_i), \quad (1.1)$$

де γ – число менше 1, яке називається коефіцієнтом навчання, t – поточний крок.

Залежно від розміру пакета B поділяють дві версії методу.

Якщо $B = 1$, тоді градієнтний спуск називається Стохастичним Градієнтним Спуском або SGD (від англійської назви Stochastic Gradient Descent), а коли $B > 1$ – Пакетним Градієнтним Спуском (або Mini-Batch Gradient Descent).

Основною перевагою SGD є швидка збіжність та малі вимоги до оперативної пам'яті (рис. 1.10). До недоліків можна віднести часті зміни напрямку пошуку, тобто хаотичність чи стохастичність оновлень. Звідки метод і отримав свою назву.

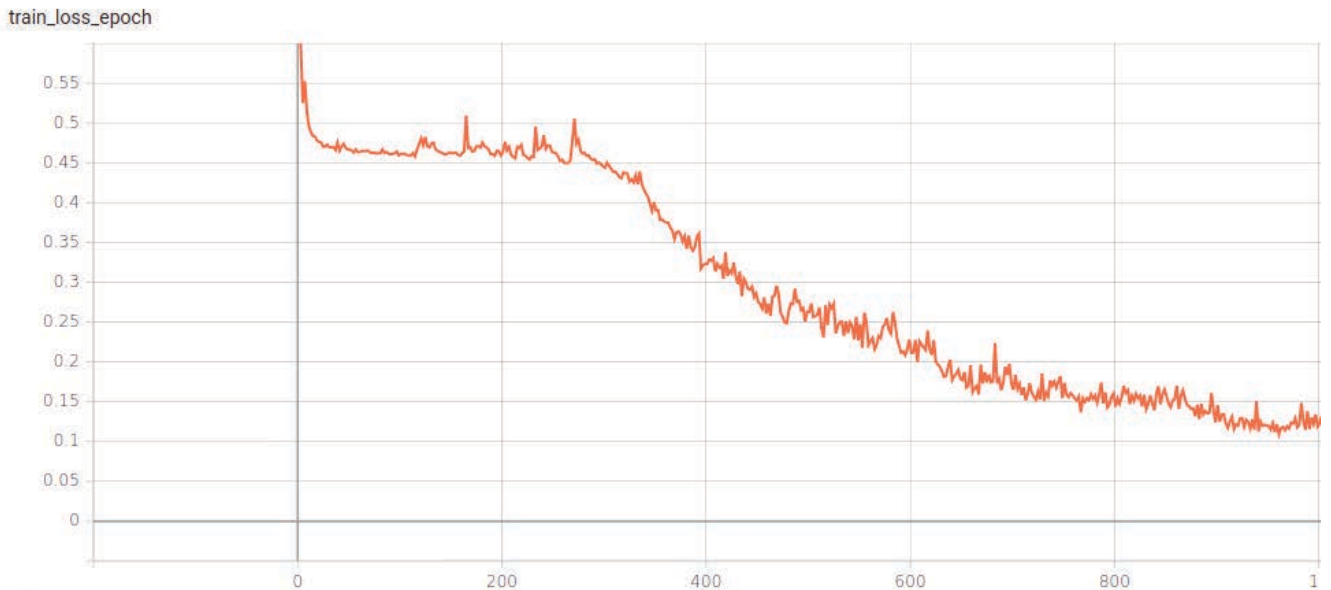


Рисунок 1.10: Графік залежності функції помилки від кількості епох для SGD

Пакетний режим вирішує деякі проблеми з SGD за рахунок менш частих оновлень напряму пошуку, які відбуваються після спроби прогнозувати результати B . Звідки випливає, що він менш хаотичний, але використовує більше пам'яті. З іншого боку, цей метод кращий для роботи на сучасних графічних прискорювачах, які мають невеликий (порівняно з оперативною пам'яттю комп'ютера) обсяг пам'яті та повільний шинний інтерфейс з CPU. У цьому випадку найефективніший метод роботи полягає у передачі максимального обсягу інформації між пам'яттю CPU та пам'яттю на GPU, після якої GPU може проводити обчислення незалежно та максимально використовувати свою архітектуру паралельних обчислень.

Розглянемо основні недоліки цих двох методів. Обидва вимагають підбору коефіцієнта γ . Дуже невелике значення даного коефіцієнта на початку навчання призводить до повільної збіжності, а дуже велике не дозволить наблизитися до мінімуму. Зазвичай, на початку тренування цей параметр вибирається досить великим, а при наближенні до локального мінімуму його зменшують. Так само має

велике значення вибір параметра B , який не тільки обмежений можливостями GPU або розмірами оперативної пам'яті комп'ютера, але він контролює частоту обчислення градієнта, а отже і стохастичність наближення до мінімуму.

Щоб зменшити стохастичність SGD та прискорити його збіжність, була розроблена його модифікація з *моментом* [77]. В її основі лежить метод, який дозволяє прискоритися в напрямку зменшення значення функції, що оптимізується. Для цього додано ще один гіпер-параметр ν , який називається моментом і відповідає за пріоритет останнього значення швидкості зміни градієнта до попередніх значень. Новий градієнтний спуск із моментом можна записати в наступному вигляді:

$$V_t = \nu V_{t-1} + \frac{1}{B} \gamma \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta), \bar{y}_i))$$

$$\Theta_{t+1} = \Theta_t - V_t,$$

де t – поточний крок. Пропоноване значення ν дорівнює 0.9 або 0.99.

Крім прискорення швидкості збіжності даний метод має лише один істотний недолік, а саме – наявність додаткового параметра ν .

Введення моменту дозволило наближатися швидше до мінімуму, але така швидкість, що все збільшується, має і недолік – на високій швидкості можна "проскочити" точку мінімуму.

Наступна модифікація градієнтного спуску називається ім'ям свого творця - Нестерова [73]. Щоб компенсувати розгін, метод змінюється і починає врахову-

вати наступний крок. Запишемо кроки методу у вигляді формули

$$V_t = \nu V_{t-1} + \frac{1}{B} \gamma \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta - \nu V(t-1)), \bar{y}_i))$$

$$\Theta_{t+1} = \Theta_t - V_t.$$

Ця модифікація має менші шанси пропустити мінімальну точку, але містить два параметри конфігурації. Крім того, існує недолік методу, що проявляється, коли деякі значення градієнтів не дорівнюють нулю. Для його усунення були запропоновані інші модифікації.

Оптимізатор Adagrad [34] змінює коефіцієнт навчання кожного параметра на кожному кроці (залежить від t). Введемо позначення

$$g_{t,i} = \frac{1}{B} \nabla_{\Theta} \sum_{i=1}^B L(f(\bar{x}_i; \Theta_{t,i}))$$

Тоді оновлення параметрів нейронної мережі на кроці $t+1$ проводитиметься за формулою

$$\Theta_{t+1,i} = \Theta_{t,i} - \frac{\gamma}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i},$$

де $G_{t,ii}$ – змінна, у якій зберігаються суми квадратів всіх градієнтів для параметра Θ_i з початку тренування до часу t ; ϵ – параметр, доданий для того, щоб уникнути поділу на 0, оскільки в процесі наближення до мінімуму $G_{t,ii}$ може виявитися дуже невеликим числом.

Таким чином, Adagrad дозволяє оновлювати більше параметрів, що мало змінюються, і робити коефіцієнт навчання невеликим для часто змінюваних параметрів. Звідси впливає один з основних недоліків цього методу, а саме – зменшення коефіцієнтів навчання зі збільшенням часу [34].

Щоб поліпшити ситуацію з можливим швидким зменшенням коефіцієнта навчання зі збільшенням часу навчання t , була створена наступна модифікація - *AdaDelta*. Її ідея полягає у збереженні суми всіх градієнтів у квадраті для деякого вікна фіксованого розміру w [70].

І нарешті, розглянемо останній метод, який містить у собі найкращі можливості всіх розглянутих модифікацій градієнтного спуску. Називається він *Adam* або *Adaptive Moment Estimation*, який працює із переміщеннями першого та другого порядків. Цей метод намагається зменшувати прискорення, якщо швидкість руху стає високою для того, щоб уникати пробігання повз точки мінімуму. На додаток до цього Adam зберігає значення середніх градієнтів, що експоненційно зменшуються, як у *AdaDelta* [18].

Спочатку обчислюються середні попередніх значень градієнтів та їхніх квадратів за формулами:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2,$$

де β_1 та β_2 – параметри із значеннями близькими до 1.

З цих формул випливає, що m_t наближає середнє значення градієнта, а v_t - нецентровану дисперсію градієнта.

Оскільки початкові значення m_t та v_t рівні 0, їхні перші значення близькі нулю й у компенсації цього застосовуються додаткові коригування

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}.$$

Внаслідок чого оновлення параметра схоже на метод *AdaDelta*

$$\Theta_{t+1} = \Theta_t - \frac{\gamma}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t,$$

де пропонувані значення для β_1 та β_2 рівні 0.9 та 0.999, відповідно, а $\epsilon = 10^{-8}$.

Існують інші модифікації, розглянутих алгоритмів: Rprop [78], ASGD [76].

Усім методам властивий в умовах багатоцільової, мультимодальної та зашумленої функції загальний недолік – «повільна» збіжність, яка суттєво збільшує час навчання. Для його усунення запропоновано модифікацію методів навчання НМ на основі градієнтного спуску, що ґрунтується на застосуванні теорії нелінійної динаміки та розроблено відповідні теоретичні положення для реалізації методів.

1.3 Подання методу градієнтного спуску як дискретної динамічної системи

Введемо поняття *динамічної системи*. Динамічна система – це математична модель деякої системи, що змінюється у часі. Умова зміни системи в часі не виключає випадку, коли система залишається у стабільному стані, який називають стаціонарним станом або еквілібріумом.

Стан системи можна представити у вигляді кінцевого числа n вимірних величин, званих ”змінними станами”. Ці числа виражаються через набір $\bar{x}_i \in \mathbb{R}$, $i = 1, \dots, n$ дійсних змінних.

Стан динамічної системи можна представити у вигляді вектора $\bar{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, тобто точкою у багатовимірному просторі. Це уявлення дозволяє використовувати геометричну мову в процесі опису змін станів динамічної системи у часі. Отже поведінка деякої одновимірної динамічної системи буде описувати-

ся поведінкою точки на прямій, – двовимірної поведінкою точки на площині, а тривимірної – у просторі.

Важливо зауважити, що безліч станів динамічної системи не обов’язково має заповнити весь простір $\mathbb{R}^n$, пов’язано це з фізичними обмеженнями. Наприклад, вартість цінних паперів може бути від’ємною. Відтак визначення простору станів можна сформулювати наступним чином.

Означення 1.1. *Простір станів (або фазовий простір) $M \subseteq \mathbb{R}$ – це множина допустимих значень змінних станів.*

Оскільки динамічна система змінюється у часі, то її змінні стани є функціями часу $x_i = x_i(t)$, $i = 1, \dots, n$, де t може бути дійсним числом (безперервний час) або натуральним числом (дискретний час). Дискретність часу не означає, що інтервали часу завжди однакові, а тільки те, що виміри проводяться через деякі інтервали. Ціна цінних паперів зазвичай доступна через різні інтервали часу, а значить, є дискретною динамічною системою.

Математично різницю між типами систем можна описати так: $x_i : \mathbb{R} \rightarrow \mathbb{R}$ (безперервна) або $x_i : \mathbb{N} \rightarrow \mathbb{R}$ (дискретна).

Незалежно від типу динамічної системи основним її завданням є визначення стану системи під час $t \neq t_0$, якщо відомий стан у деякий початковий момент часу t_0 . Це еквівалентно знанню оператора еволюції

$$\mathbf{x}(t) = \mathbf{G}(t, \mathbf{x}(t_0)), \quad (1.2)$$

де $\mathbf{x}(t) = (x_1(t), \dots, x_n(t)) \in M \subseteq \mathbb{R}^n$, а функція $\mathbf{G}(\cdot) = (G_1(\cdot), \dots, G_n(\cdot)) : M \rightarrow M$. Якщо знаємо цей *еволюційний оператор* $\mathbf{G}$, тоді є можливість дізнатися про стан системи в будь-який майбутній момент часу $t > t_0$, а також дізнатися про

стан системи в минулому при $t < t_0$. Хоча у більшості практичних моделей інтерес полягає у розумінні поведінки системи в далекому майбутньому і особливо при $t \rightarrow +\infty$.

Зазначимо, що множина операторів $\{G_t\}$ утворює групу щодо операції суперпозиції в тому сенсі, що суперпозиція операторів $G_{t_1} \oplus G_{t_2}$ визначає оператор $G_{t_1+t_2} \in \{G_t\}$, G_0 – нейтральний елемент, G_{-t} – зворотний оператор.

Вектор-функція $\mathbf{x}(t)$, визначена за формулою (1.2), являє собою параметричну криву в n -мірному просторі, тобто *траєкторією*. У безперервному випадку траєкторія є кривою, а в дискретному – лічильною безліччю точок.

Положення рівноваги – це траєкторія системи $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$, в якій усі змінні стани є постійними

$$\mathbf{x}(t) = \mathbf{G}(t, \mathbf{x}^*) = \mathbf{x}^* \text{ при } t > t_0.$$

Стаціонарна точка може бути точкою тяжіння. Це означає, що будь-яка траєкторія, що проходить через неї залишається у цій точці назавжди. Поняття точки тяжіння можна поширити на підмножини.

Означення 1.2. Множина $A \subseteq M$ називається *притягуючою*, якщо з $\mathbf{x}(t_0) \in A$ слід $\mathbf{x}(t) = \mathbf{G}(t, \mathbf{x}(t_0)) \in A$ для будь-якого $t > t_0$.

Отже будь-яка траєкторія, що стартує всередині множини, що притягує, не може вийти за її межі. На основі цього можна запровадити поняття інваріантної множини.

Означення 1.3. Замкнена множина $A \subseteq M$ називається *інваріантною*, якщо $\mathbf{G}(t, A) = A$. Тобто будь-яка підмножина $A \subset A$ не є притягуючою.

У визначенні інваріантної множини використовувалося позначення

$$\mathbf{G}(t, A) = \{\mathbf{x}(t) \in M : \exists t \geq t_0 \text{ і та } \mathbf{x}(t_0) \in A \text{ такі що } \mathbf{x}(t) = \mathbf{G}(t, \mathbf{x}(t_0))\}.$$

Особливий інтерес представляє питання про поведінку траєкторій, які починаються поблизу інваріантних множин. Цікаво, чи траєкторія динамічної системи залишатиметься поблизу інваріантної множини, як довго буде існувати і чи не піде нескінченно далеко. Випадок, коли траєкторія залишається на окілі визначається через стійкість за Ляпуновим.

Означення 1.4. (стійкість за Ляпуновим) *Інваріантна множина A називається стійкою, якщо для кожної окіла U існує інший окіл V множини A , де $V \subseteq U$, таке що будь-яка траєкторія, що починається в V , залишається всередині множини U .*

Це визначення описує траєкторії, які починаються поблизу інваріантної множини і назавжди залишаються в околі цієї множини. Нестійкість визначається як зворотне до стійкості.

Означення 1.5. *Інваріантна множина A називається асимптотично стійкою якщо:*

1) *A стійке за Ляпуновим*

2) $\lim_{t \rightarrow +\infty} \mathbf{G}(t, \mathbf{x}) \in A$ *для будь-якого початкового $\mathbf{x} \in V$.*

При цьому друга умова не є сильнішою, ніж перша, тому що можливий випадок, коли траєкторія починається поблизу інваріантної множини A і віддаляється від неї на велику відстань. І лише після деякого часу починає асимптотичне наближення до A . Так що перша умова останнього визначення не дозволяє траєкторії віддалятися від A .

У дискретному випадку зручніше користуватися не оператором еволюції (який може не мати представлення у вигляді кінцевої формули), а функцією, яка описує закон переходу системи з одного стану до іншого для кожного кроку. Звичайно, знаючи цю функцію, можна записати оператор еволюції та навпаки.

Нехай задана функція $f(x)$, яка визначена і безперервна на множини M , може бути як обмеженою так і необмеженою підмножиною $\mathbb{R}^N$, а також може збігатися з ним. Додатково потрібно, щоб множина значень функції $f(x)$ так само була підмножиною множини M .

Розглянемо динамічну систему, стан якої змінюється через дискретні проміжки часу, отриману з урахуванням функції $f(x)$

$$x_{n+1} = f(x_n), \quad (1.3)$$

де $n = 0, 1, \dots$, и $x_n \in M \subset \mathbb{R}^N$. Цю саму систему можна записати в іншому (іноді більш зручному вигляді)

$$x \rightarrow f(x), x \in M \subset \mathbb{R}^N.$$

Множину траєкторій дискретних динамічних систем можна розбити на три види: неперіодичні рішення, цикли та нерухомі точки. Періодичне рішення – це рішення, коли в якийсь момент часу t_k стан системи повторюється. Це гарантує появу циклу та повторюваність послідовності станів. І остання можливість – це нерухома точка $x_0 = f(x_0)$.

Розглянемо докладніше періодичний (циклічний) випадок.

Нехай p є деяким натуральним числом. Розглянемо послідовність станів

$$x_0^*, x_1^*, x_2^*, \dots, x_p^*,$$

яка є циклом періоду p , якщо

$$x_1^* = f(x_0^*), x_2^* = f(x_1^*), x_3^* = f(x_2^*), \dots, x_{p-1}^* = f(x_{p-2}^*), x_0^* = f(x_{p-1}^*).$$

Зрозуміло, що цикли повторюватимуться нескінченно і за їхню початкову точку x_o^* можна взяти будь яку точку самого p -циклу. І легко довести таке твердження.

Лема 1.1. *Якщо система має p -цикл, тоді будь-яке рішення системи є kp -циклом для будь-якого $k \in \mathbb{N}$.*

З цього твердження випливає, що разом з одним циклом існує лічена множина інших циклів кратного розміру. Надалі за довжину циклу братимемо мінімальний елемент.

Важливо відзначити, що система не завжди починається з нерухомої або циклічної точки, а може переходити у ці стани після деякого переходу.

Для циклів поняття стійкості за Ляпуновим можна задати за аналогією через композицію функції. Нехай система (1.3) має p -цикл $x_0^*, x_1^*, \dots, x_{p-1}^*$. Тоді будь-який із векторів цього циклу є рішенням рівняння

$$x_{n+1} = f^{(p)}(x_n), n = 0, 1, 2, \dots,$$

де $f^{(p)} = f \circ f \circ f \circ f \circ \dots \circ f$ (p раз) – композиція функції f .

Класифікуємо множину динамічних систем відповідно до їх властивостей. Спочатку розділимо системи простору на *кінцевомірні* та *нескінченномірні* простори. Цей поділ відбувається на основі розмірності простору станів системи Q .

Якщо простір станів є кінцевим лінійним простором, тоді така система називається *кінцевомірною*. В іншому випадку - *нескінченномірною*.

Іншою основною характеристикою динамічних систем є їх дискретність або безперервність. Якщо стан системи залежать безперервно від часу t , тоді система називається *безперервною*, а якщо стан системи змінюється тільки при дискретних значеннях часу - дискретною динамічною системою.

Введемо поняття *стискаючого відображення*. Для цього розглянемо рівняння

$$x = f(x),$$

де функція $f(x)$ має своєю областю визначення та значення $\mathbb{R}^N$. Таким чином на цю функцію можна дивитися як на відображення $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$.

Основний метод вирішення зазначеного рівняння ґрунтується на *принципі стискаючих відображень*, який у свою чергу використовує ще одне поняття *нерухомої* точки функції f . Такою точкою називається така точка x^* для якої $f(x^*) = x^*$.

Означення 1.6 (Інваріантна множина). Множину $M \subset \mathbb{R}^N$ називатиме *інваріантною* для функції f , якщо $f(M) \subseteq M$.

У випадку, коли в $\mathbb{R}^N$ задана метрика $\| \cdot \|$ і на інваріантній для f множині виконується нерівність

$$\|f(x) - f(y)\| \leq q\|x - y\|,$$

$q < 1$, для $\forall x, y \in M$. Тоді функцію f називають *стискаючою* на множині M , а коефіцієнт q – *коефіцієнтом стиснення*.

Подамо (1.1) у вигляді дискретної динамічної системи, основні поняття про

яку будуть описані в другому розділі. Для цього введемо додаткове позначення

$$G(\vartheta) = \vartheta - \frac{1}{B} \gamma \nabla_{\vartheta} \sum_{i=1}^B L(f(\bar{x}_i; \vartheta), \bar{y}_i).$$

У цих позначках (1.1) можна подати у вигляді

$$\Theta_{t+1} = G(\Theta_t), \quad (1.4)$$

що і є дискретною динамічною системою.

Нехай ця дискретна система має стаціонарну точку (або цикл довжини 1) Θ^* , тоді $\Theta^* = G(\Theta^*)$. Це означає, що для цієї точки

$$\nabla_{\Theta} \frac{1}{B} \sum_{i=1}^B L(f(\bar{x}_i; \Theta^*), \bar{y}_i) = 0.$$

Отже, кожна стаціонарна точка побудованої дискретної динамічної системи одночасно є точкою мінімуму для середньої помилки щодо підмножини навчальних даних.

Зводячи задачу навчання до пошуку стаціонарних точок дискретних динамічних систем, можна скористатися розробленими методами стабілізації циклів довжини T , що розроблені у статті [27]. Застосовуючи формули (1.1) та (1.4) пропонується модифікація методу градієнтного спуску з урахуванням зворотного зв'язку з запізненням в ДДС (почалося з робіт [26, 29, 77]):

$$\Theta_{t+1} = (1 - \lambda) G \left(\sum_{j=1}^N a_j \Theta_{n-jT+T} \right) + \lambda \sum_{j=1}^N b_j \Theta_{n-jT+1}, \quad (1.5)$$

де параметри a_j і b_j підпорядковуються умові $\sum_j a_j = \sum_j b_j = 1, \forall \lambda \in \mathbb{R}$.

У разі використання прогнозуючого зворотного зв'язку в ДДС, а також фор-

мулами (1.1) та (1.4):

$$\Theta_{t+1} = G(\Theta_t) + u_t, \quad (1.6)$$

де зворотний зв'язок виявляється у такому вигляді $u_t = (a_1 - 1)G(\Theta_t) + \sum_{j=2}^N a_j G^{((j-1)T+1)}(\Theta_t)$, $G^{(1)}(\Theta) = G(\Theta)$, $G^{(k)}(\Theta) = G(G^{(k-1)}(\Theta))$, $\sum_{j=1}^N a_j = 1$, $k \in \mathbb{N}$.

Графіки градієнтного спуску з моментом, градієнтного спуску за Нестеровим та ДДС зі зворотною зв'язком (1.5) представлені на рисунку (1.11). З графі-

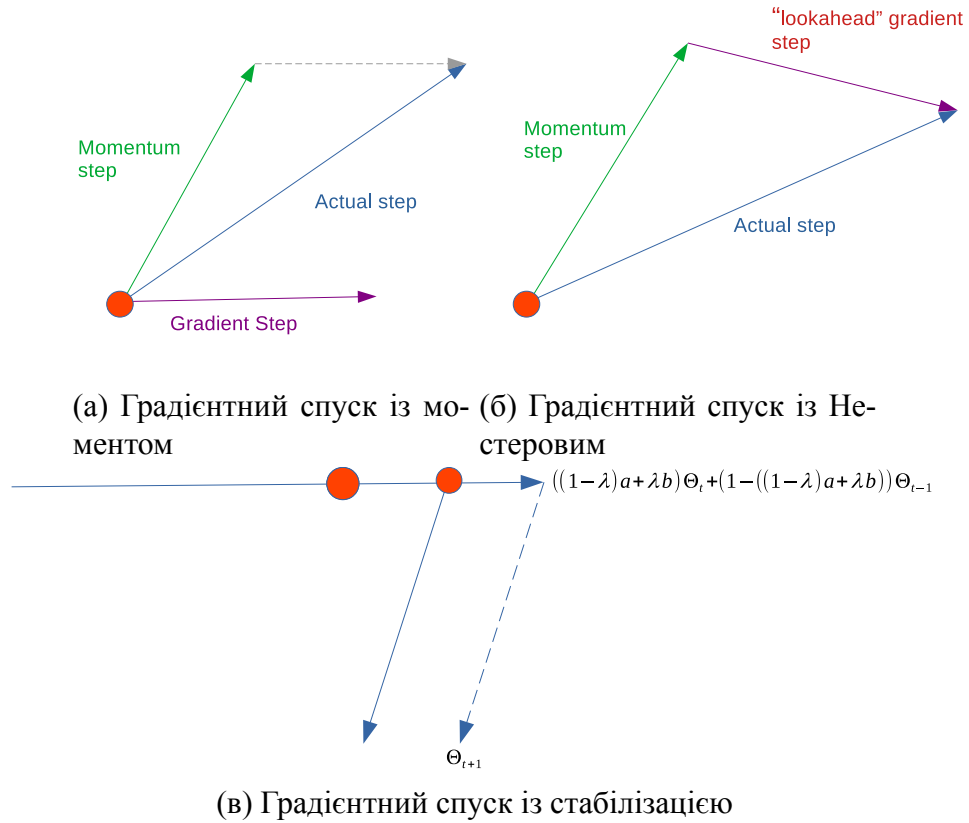


Рисунок 1.11: Порівняння градієнтного спуску з моментом, градієнтного спуску за Нестеровим та ДДС зі зворотною зв'язком (1.5)

ків видно, що три модифікації градієнтного спуску сильно відрізняються одна від одної і більша кількість параметрів може дозволити точніше керувати поведінкою методу (1.5) у різних ситуаціях.

Питання про те, як вибирати параметри для (1.5) та (1.6) та їх математичні властивості будуть розглянуті у другому розділі.

1.4 Висновки до першого розділу

У першому розділі дисертаційної роботи досліджені особливості застосування нейронних мереж для сегментації та класифікації об'єктів на зображеннях. Розглянуто різні архітектури глибоких мереж загального та спеціального призначення.

Показано, що ускладнення нейронних мереж та збільшення їхньої глибини безпосередньо пов'язано зі збільшенням обсягу даних, необхідних у процесі навчання, а також часу на процес навчання. Це призвело до 300 тис. кратного збільшення витрачених ресурсів на навчання глибоких нейронних мереж.

Показано, що навчання глибоких нейронних мереж проводитиметься різними модифікаціями градієнтного спуску, який можна подати у вигляді дискретної динамічної системи, що дозволяє використовувати методи нелінійної динаміки.

Також, удосконалено метод градієнтного спуску на основі положень нелінійної динаміки (**перший пункт новизни наукового дослідження**), що дозволило розробити модифікації методу з урахуванням зворотнього зв'язку з запізненням та прогнозуючого зворотнього зв'язку в ДДС.

РОЗДІЛ 2

МЕТОД ПОШУКУ ПАРАМЕТРІВ ДЛЯ ЗАПРОПОНОВАНИХ МОДИФІКАЦІЙ МЕТОДУ ГРАДІЄНТНОГО СПУСКУ

2.1 Зв'язок задачі пошуку та стабілізації періодичних точок із задачами геометричного комплексного аналізу

Розглянемо динамічну систему (1.3), яка має деякий нестійкий T -цикл $(x_1^*, \dots, x_T^*)$. Визначимо мультиплікатори $\mu_1, \dots, \mu_m$ циклу як нулі характеристичного поліному

$$\det \left(\mu I - \prod_{j=1}^T Df(x_{T-j+1}^*) \right) = 0,$$

де $Df(x_j^*)$ матриця Якобі вектор-функції $f(x)$, обчисленої у точках циклу x_j^* .

Можна показати, що T -цикли системи (1.3) також є циклами системи (1.5).

Для збереження інваріантної опуклості множини A системи (1.5) необхідні додаткові умови: $a_j \in [0, 1]$, $b_j \in [0, 1]$.

Треба знайти такі значення параметрів a_j і b_j , щоб вони задовольняли заданим обмеженням, так що T -цикл системи (1.5) був локально асимптотично стійким, а величина N була найменшою.

Якщо в (1.5) покласти $\gamma = 0$, то отримаємо систему з нелінійним зворотним зв'язком. Якщо ж у (1.5) покласти $N = 1$, то $a_1 = b_1 = 1$, отже, отримаємо напівлінійний зворотній зв'язок. З цього випливає, що система (1.5) містить як нелінійний, так і напівлінійний контроль.

Дослідження стійкості T -циклу системи (1.5) починається з виведення характеристичного рівняння, де класичними методами через побудову матриці Якобі спеціального відображення в околі циклу [72], знаходиться характеристичний поліном цієї матриці. Цей поліном має громіздкий і незручний вигляд для спро-

щення та аналізу.

У роботах [25, 80] характеристичний поліном був побудований виходячи з іншого відображення, так що він має вигляд

$$f(z) = \prod_{j=1}^m \left([1 - \gamma p(z)]^T - (1 - \gamma)^T \mu_j z [q(z)]^T \right),$$

де $q(z) = a_1 + a_2 z + \dots + a_N z^{N-1}$ та $p(z) = b_1 + b_2 z^{N-1} + \dots + b_N z^N$ з нормалізацією $q(1) = p(1) = 1$.

Отриманий поліном виявився зручнішим для досліджень та в [80] доведено еквівалентність класичного методу та альтернативного методу виводу характеристичного полінома.

Наступний крок дослідження стійкості циклів полягає в аналізі розташування нулів нового характеристичного полінома на комплексній площині. Локальна стійкість циклів різницевого схем еквівалентна стійкості по Шуру характеристичного полінома, що відповідає цьому циклу. Це твердження можна подати у вигляді наступної лема.

Лема 2.1. *T-цикл системи (1.5) локально асимптотично стійкий тоді і лише тоді, коли всі нулі характеристичного полінома лежать у відкритому центральному одиничному колі $D = \{z : |z| < 1\}$ [97].*

Однак застосувати відомі критерії стійкості по Шуру для характеристичного полінома не вдається, тому що величини мультиплікаторів $\mu_1, \dots, \mu_m$ невідомі. Тому для перевірки локальної стійкості циклів системи (1.5) застосуємо геометричний критерій стійкості.

$$\text{Позначимо } \Phi(z) = (1 - \gamma)^T \frac{z(q(z))^T}{(1 - \gamma p(z))^T}.$$

Лема 2.2. *Усі нулі характеристичного полінома лежать у відкритому централь-*

ному колі D тоді і тільки тоді, коли справедливі включення

$$\mu_j \in (\overline{\mathbb{C}} \setminus \Phi(\overline{\mathbb{D}}))^*, j = 1, \dots, m, \quad (2.1)$$

де $\overline{\mathbb{D}}$ – замкнене центральне одиничне коло, $\overline{\mathbb{C}}$ – розширена комплексна площина, зірочкою позначена операція інверсії: $(z)^* = \frac{1}{z}$.

Мультиплікатори циклу невідомі, отже, T -цикл буде локально асимптотично стійкий, якщо множина $(\overline{\mathbb{C}} \setminus \Phi(\overline{\mathbb{D}}))^*$ покриває множину M локалізації мультиплікаторів. Це означає, що множина $(\overline{M})^*$ має бути виключною для образу кола $\mathbb{D}$ при відображенні $\Phi(z)$. Ця властивість буде основним для побудови коефіцієнтів управління $a_1, \dots, a_N, b_1, \dots, b_N$.

Таким чином наше завдання звелось до задачі пошуку функції Φ , яка залежить від двох поліномів $q(z)$ і $p(z)$ і параметра $\gamma \in [0, 1)$, причому $q(1) = 1$, $p(0) = 0$, $p(1) = 1$, отже, $\Phi(0) = 0$, $\Phi(1) = 1$.

Накладемо додаткові обмеження на мультиплікатори, тобто припустимо, що мультиплікатори мають від'ємну дійсну частину. Будемо розглядати два випадки:

Випадок А: $\{\mu_1, \dots, \mu_m\} \in \{\mu \in \mathbb{R} : \mu \in (-\hat{\mu}, 1)\}$.

Випадок Б: $\{\mu_1, \dots, \mu_m\} \in \{\mu \in \mathbb{C} : |\mu + R| < R\} \cup \mathbb{D}$, де $\mathbb{D} = \{z : |z| < 1\}$.

Геометричний критерій стійкості, разом із накладеними обмеженнями, зводить завдання до кількох оптимізаційних проблем, вирішення яких вимагають знаходження необхідних для стабілізації коефіцієнтів.

Досліджуємо випадок, коли $\gamma = 0$ у формулі (1.5). Для випадку А отримаємо оптимізаційну задачу виду

$$I_N^{(T)} = \sup_{a_j, b_j} \min_{t \in [0, \pi 0]} \{ \Re(\Phi(e^{it})) : \Im(\Phi(e^{it})) = 0 \}, \quad (2.2)$$

випадок Б призводить до іншої задачі оптимізації

$$J_N^{(T)} = \sup_{a_j, b_j} \min_{t \in [0, \pi 0]} \{ \Re (\Phi (e^{it})) \}. \quad (2.3)$$

Використовуючи наведені вище оптимізаційні задачі, аналітичний запис геометричного критерію стабільності може бути записаний у вигляді:

Лема 2.3. Система (1.5) має стійкий T -цикл якщо

$$\hat{\mu} \left| I_N^{(T)} \right| < 1 \text{ (випадок А)} \quad (2.4)$$

i

$$R \left(2 \left| J_N^{(T)} \right| \right) \leq 1 \text{ (випадок Б)}. \quad (2.5)$$

Оптимізаційна задача знаходження $I_N^{(T)}$ було вирішене для $T = 1, 2$ методами гармонічного аналізу в [28, 29, 30]. У загальному випадку пошук оптимального значення для $I_N^{(T)}$ и $J_N^{(T)}$ буде проводитись у класі поліномів, які можуть бути зручними кандидатами для вирішення екстремальної задачі.

2.2 Оптимальні многочлени та їх властивості

Вперше результати, надані за цим розділом, було оголошено у переддруку [31].

Нехай $N > 1$ є цілим числом, і далі вважатимемо $F_N(z)$ позначенням поліному $F_N(z) = z + \sum_{j=2}^N a_j z^j$ з дійсними коефіцієнтами.

Досліджуємо наступну задачу оптимізації.

Задача 1. Знайти

$$J_N = \sup_{a_2, \dots, a_N \in \mathbb{R}} \left(\inf_{z \in \mathbb{D}} \{ \Re(F_N(z)) : \Im(F_N(z)) = 0 \} \right)$$

і знайти екстремальний поліном $F_N(z)$.

Зрозуміло, що $\Im(F_N(e^{it})) = 0$ коли $t = 0$ або $t = \pi$.

Нижче покажемо, що

$$J_N = -\frac{1}{4} \sec^2 \frac{\pi}{N+2}.$$

Екстремальні поліноми є однозначно визначеними

$$\frac{1}{U'_N\left(\cos \frac{\pi}{N+2}\right)} \sum_{j=1}^N U'_{N-j+1}\left(\cos \frac{\pi}{N+2}\right) U_{j-1}\left(\cos \frac{\pi}{N+2}\right) z^j, \quad (2.6)$$

де $U_j(x)$ – поліноми Чебишева другого роду, а $U'_j(x)$ – похідна від $U_j(x)$.

Рішення та поліноми було опубліковано в [7, 31].

Задача 1 пов'язана з оцінкою радіусу Кебе для поліномів.

Стисло наведемо відповідну концепцію. Зображення кола одиничного радіусу може мати отвори, тому, при бажанні стиснути зображення вздовж дійсної осі, потенційно існують придатні для цього незаповнені місця. Отже, можна очікувати, що екстремальний поліноміальний образ кола не має порожніх просторів (дірок). Виявляється, наведена вище евристика дійсно має рацію.

Проілюструємо доказ у найпростішому випадку $N = 2$: $p(z) = z + az^2$.

Без втрати загальності можна вважати $a > 0$. Тоді

$$p(e^{it}) = \cos t + a(2 \cos^2 t - 1) + i \sin t(1 + 2a \cos t) =$$

$$\cos t(1 + 2a \cos t) - a + i \sin t(1 + 2a \cos t) =: C(t) + iS(t).$$

Тож, якщо $S(t_0) = 0$ тоді $1 + 2a \cos t_0 = 0$ й $\cos t_0 = -\frac{1}{2a}$, якщо $0.5 \leq a$.

Далі, $C(t) = -a + 2a((2a)^{-1} + \cos t) \cos t$ отримуємо наступну викладку

$$C(t) = -a + 2a(\cos t - \cos t_0) \cos t, \quad (2.7)$$

$$S(t) = 2a(\cos t - \cos t_0) \sin t. \quad (2.8)$$

Припустимо, що $C(t_0)$ є найбільшим значенням для $0.5 \leq a$. Можемо показати, що за допомогою варіаційного методу можна отримати поліном з більшим значенням. Для цього змінюємо значення t_0 , тобто введемо поліноми

$$C_\theta(t) = N(\theta)(-a + 2a(\cos t - \cos \theta) \cos t), \quad (2.9)$$

$$S_\theta(t) = N(\theta)(2a(\cos t - \cos \theta) \sin t). \quad (2.10)$$

Оскільки новий поліном $p_\theta(z)$ потрібно нормувати, коефіцієнт для $\sin t$ має складати 1. Отже, $-N(\theta)2a \cos \theta = 1$, або $N(\theta) = -\frac{1}{2a \cos \theta}$. Тепер $C_\theta(\theta) = \frac{1}{2 \cos \theta} > \frac{1}{2 \cos t_0} = C(t_0)$, якщо $\pi/2 < t_0 < \theta < \pi$.

Таким чином, якщо існує такий поліном $p_0(z)$, що є оптимальним для задачі

$$J := \max_a \min_t \{C(t) : S(t) = 0\}, \quad (2.11)$$

тоді мінімум досягається при $t_0 = \pi$. Наблизились до задачі

$$J_2 = \max_a \min_t \{C(t) : S(t) = 0, S(t) > 0 \text{ for } 0 < t < \pi\} =$$

$$\max_a \{C(\pi) : \sin t(1 + 2a \cos t) > 0, \text{ for } 0 < t < \pi\} \quad (2.12)$$

або

$$J_2 = \max_a \{a - 1 : 1 + 2a \cos t > 0 \text{ for } 0 < t < \pi\}.$$

Зрозуміло, що найбільше значення a таке, що $1 + 2a \cos t > 0$ при $0 < t < \pi$, дорівнює $1/2$, отже

$$J_2 = -\frac{1}{2}. \quad (2.13)$$

Цю ситуацію добре ілюструють рисунки 2-4 вище.

Загальний випадок в принципі має аналогічний вигляд, але технічно є набагато складнішим.

А саме, подання у формі (2.7), (2.8) розширено для загального випадку у лемі 2.4 нижче. З застосуванням варіаційних методів, співвідношення (2.9), (2.10) використовуються для доведення того, що не існує кореня дійсної частини, крім нуля та π (лема 2.9 нижче), що призводив би до загального співвідношення, подібного до (2.12). Розкладаючи синус на множники, прийшли до задачі максимізації деякої величини за умови невід'ємності певного косинусного полінома. Для вирішення цієї задачі застосовано представлення невід'ємного тригонометричного полінома за Фейєром-Рісом, що дозволяє звести задачу до питання оптимізації за типом задачі Рейлі (теорема 1). Розв'язання цієї задачі полягає в обчисленні узагальнених власних значень, тобто нулей визначника розміру $N \times N$ деякої 5-діагональної матриці. Ця частина є найбільш складною з точки зору обчислень (лема 2.6). Інший проблемний аспект полягає в обчисленні узагальнених власних векторів (леми 2.5, 2.7 та 2.10), тобто, доведенні єдиності екстремального поліному.

Наступна лема керована принципом чистого гармонічного аналізу і складає

самостійний предмет зацікавленості.

Лема 2.4. *Нехай*

$$C(t) = \sum_{j=1}^N a_j \cos jt, \quad S(t) = \sum_{j=1}^N a_j \sin jt \quad (2.14)$$

є парою супряжених тригонометричних поліномів з дійсними коефіцієнтами. Припустимо, що

$$S(t_1) = \dots = S(t_m) = 0, \quad C(t_1) = \dots = C(t_m) = \gamma, \quad (2.15)$$

для деякого $t_1, \dots, t_m \in (0, \pi)$ з $2m \leq n$. Тоді тригонометричні поліноми (2.14) мають вигляд

$$C(t) = \gamma + \prod_{k=1}^m (\cos t - \cos t_k) \cdot \sum_{j=m}^{N-m} \alpha_j \cos jt,$$

$$S(t) = \prod_{k=1}^m (\cos t - \cos t_k) \cdot \sum_{j=m}^{N-m} \alpha_j \sin jt,$$

де коефіцієнти $\alpha_m, \dots, \alpha_{N-m}$ однозначно визначаються $\gamma, a_1, \dots, a_N$, та $\alpha_m = -2^m \gamma$, $a_1 = \frac{\alpha_{m+1}}{2^m} - \frac{\alpha_m}{2^{m-1}} \sum_{j=1}^m \cos t_j$.

Перш ніж навести доказ, розглянемо кілька прикладів. Нехай $N = 2n$, і

$$C(t) = \sum_{j=1}^N a_j \cos jt, \quad S(t) = \sum_{j=1}^N a_j \sin jt,$$

тож припустимо, що $a_j = 1$ для j парних і $a_j = 0$ для j непарних, $j = 1, \dots, N$.

Тоді

$$C(t) = \sum_{j=1}^n \cos 2jt = \frac{\sin nt \cdot \cos(n+1)t}{\sin t},$$

$$S(t) = \sum_{j=1}^n \sin 2jt = \frac{\sin nt \cdot \sin(n+1)t}{\sin t}.$$

Зрозуміло, що $S(t_k) = 0$ і $C(t_k) = -1$ для $t_k = \frac{\pi k}{n+1}$, $k = 1, \dots, n$.

Тепер застосуємо лему 2.4. У наших позначеннях $m = n$ і $\gamma = -1$, отже, $\alpha_m = -\gamma 2^m = 2^n$, таким чином

$$C(t) = \sum_{j=1}^N a_j \cos jt = -1 + \prod_{k=1}^m (\cos t - \cos t_k) \cdot \sum_{j=m}^{N-m} \alpha_j \cos jt,$$

прийшли до наступного вдалого співвідношення

$$\sum_{j=1}^n \cos 2jt = -1 + 2^n \cos nt \prod_{k=1}^n (\cos t - \cos \frac{\pi k}{n+1}).$$

Припускаючи $t = 0$, отримуємо добре відому тотожність

$$\frac{n+1}{2^n} = \prod_{k=1}^n \left(1 - \cos \frac{\pi k}{n+1}\right).$$

Отримуємо ще одне співвідношення

$$\sum_{j=1}^n \sin 2jt = 2^n \sin nt \prod_{k=1}^n (\cos t - \cos \frac{\pi k}{n+1}).$$

Подібним чином можна отримати мультиплікаційні формули для ядер Дірі-хле

$$\sum_{j=1}^{2n} \sin jt = 2^n \prod_{k=1}^n \left(\cos t - \cos \frac{2\pi k}{2n+1}\right) \sin nt,$$

$$\sum_{j=1}^{2n} \cos jt = -1 + 2^n \prod_{k=1}^n \left(\cos t - \cos \frac{2\pi k}{2n+1}\right) \cos nt,$$

$$\sum_{j=1}^{2n+1} \sin jt = 2^n \prod_{k=1}^n \left(\cos t - \cos \frac{\pi k}{n+1} \right) (\sin nt + \sin(n+1)t),$$

$$\sum_{j=1}^{2n+1} \sin jt = -1 + 2^n \prod_{k=1}^n \left(\cos t - \cos \frac{\pi k}{n+1} \right) (\cos nt + \cos(n+1)t).$$

Далі стає задача знаходження визначника 5-смугової матриці Тепліца [98].

Для початку нам знадобляться деякі властивості поліномів Чебишева другого роду.

Лема 2.5. *Дійсними є такі тотожності:*

$$U_{j-1}(x)U_{j+k}(x) = U_j(x)U_{j+k-1}(x) - U_{k-1}(x),$$

$$U_{j+k}(x) = U_j(x)U_k(x) - U_{j-1}(x)U_{k-1}(x),$$

$$\begin{aligned} \frac{d}{dx}U_k(x) &= \frac{1}{2(1-x^2)}((k+2)U_{k-1}(x) - kU_{k+1}(x)) \\ &= \frac{1}{1-x^2}((k+1)U_{k-1}(x) - kxU_k(x)). \end{aligned}$$

Доведення. Це випливає з визначення поліномів Чебишева шляхом простих обчислень.

Тепер визначимо дві матриці $N \times N$:

$$A = \begin{pmatrix} 0 & 1/2 & 0 & 0 & \dots \\ 1/2 & 0 & 1/2 & 0 & \dots \\ 0 & 1/2 & 0 & 1/2 & \dots \\ 0 & 0 & 1/2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 1/2 & 0 & \dots \\ 0 & 0 & 0 & 1/2 & \dots \\ 1/2 & 0 & 0 & 0 & \dots \\ 0 & 1/2 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}. \quad (2.16)$$

Наступна лема 2.6 також складає предмет особливої зацікавленості з кількох причин. Це ключова лема, і водночас - основна обчислювальна проблема. Як

правило, у задачах екстремальних поліномів задіяні 3-х смугові матриці. Наприклад, фундаментальним результатом є добре відоме подання поліномів Чебишева з застосуванням 3-смугової матриці Тепліца [90]

$$U_N(x) = \det \begin{pmatrix} 2x & 1 & 0 & \cdots & 0 \\ 1 & 2x & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 1 & 2x \end{pmatrix}.$$

З цієї точки зору поява поліномів Чебишева в наведеній нижче лемі не є великою несподіванкою. Однак, дещо несподіваною є поява похідної від поліномів Чебишева. Оскільки матриці (2.16) є 5-смуговими, обчислення набагато складніше, ніж у випадку 3-смугових матриць. Це може бути першим прикладом придатної до застосування формули, яка містить такі матриці.

Лема 2.6. *Нехай I означає одиничну матрицю, а матриці A і B визначаються (2.16). Тоді*

$$\det(4x^2(I - A) - (I - B)) = \frac{1}{2^{N+2}x} U_{N+1}(x) U'_{N+1}(x).$$

Наслідок 2.0.1. *Корені рівняння*

$$\det(4x^2(I - A) - (I - B)) = 0$$

є дійсними числами $\{\pm\mu_j\}_{j=1}^{\lfloor \frac{N+1}{2} \rfloor}$, $\{\pm\nu_j\}_{j=1}^{N-\lfloor \frac{N+1}{2} \rfloor}$, де $\mu_j = \cos \frac{j\pi}{N+2}$, та $U'_{N+1}(\nu_j) =$

0, які можна упорядкувати в наступний спосіб

$$0 < \mu_{\frac{N+1}{2}} < \nu_{\frac{N-1}{2}} < \dots < \nu_1 < \mu_1 \quad \text{if } N \text{ є непарний,}$$

$$0 < \nu_{\frac{N}{2}} < \mu_{\frac{N}{2}} < \dots < \nu_1 < \mu_1 \quad \text{if } N \text{ є парним.}$$

Висновок випливає з того факту, що нулі поліномів $U_{N+1}(x)$ and $U'_{N+1}(x)$ чергуються.

Наступна задача: пошук власних векторів.

Лема 2.7. Нехай A і B аналогічні поданому в попередній лемі. Сімейство з одним параметром

$$c\delta^{(0)} \left(\cos \frac{j\pi}{N+2} \right),$$

де $c \in \mathbb{R}$, $\delta^{(0)}(x) = (U_0(x)U_1(x), \dots, U_{N-1}(x)U_N(x))^T$, а T позначає транспонування, є рішенням системи лінійних рівнянь у δ ,

$$\left(4\cos^2 \frac{j\pi}{N+2} (I - A) - (I - B) \right) \delta = 0, \quad j = 1, \dots, \left\lfloor \frac{N+1}{2} \right\rfloor.$$

Лема 2.8. Область $F_N(\mathbb{D})$ містить диск радіусом $\frac{1}{N}$ з центром на нулі.

Зауваження 2.1. З теореми Вієта [97] випливає, що $\left| \frac{a_j}{\gamma} \right| \leq \binom{N}{j}$, $j = 1, \dots, N$, отже $\sum_{j=1}^N \left| \frac{a_j}{\gamma} \right| \leq 2^N - 1$ і $\sum_{j=1}^N |a_j| \leq (2^N - 1) |\gamma|$.

З рівності $J_2 = -\frac{1}{2}$ (див. (2.13)) випливає $J_N \geq -\frac{1}{2}$ зважаючи на монотонність J_N . Нехай $A = \{F_N(z) : F(z) = -1 \text{ для по } z \in \mathbb{D}\}$, тоді супремум можна обмежити множиною A , тобто

$$J_N = \sup_{F \in A} \left(\inf_{z \in \mathbb{D}} \{ \Re(F_N(z)) : \Im(F_N(z)) = 0 \} \right).$$

Оскільки -1 є винятковим значенням для $F_N(z)$ у $\mathbb{D}$, з зауваження 2.1 випливає, що $|a_j| \leq 2^N$ для $F_N \in A$ і тому

$$J_N \leq \sup_{|a_j| \leq 2^N} \left(\inf_{z \in \mathbb{D}} \{ \Re(F_N(z)) : \Im(F_N(z)) = 0 \} \right).$$

Тепер розглянемо функцію $\rho_N : \mathbb{R}^N \rightarrow \mathbb{R}$

$$\rho_N(a_2, \dots, a_N) = \min \{ C(t) : t \in T \cup \{\pi\} \},$$

де $F_N(e^{it}) = C(t) + iS(t)$ і T множина точок у $(0, \pi)$, де $S(t)$ змінює знак. Оскільки $T \cup \{\pi\}$ міститься в множині нулів $S(t)$, отримуємо

$$\sup_{|a_j| \leq 2^N} \left(\inf_{z \in \mathbb{D}} \{ \Re(F_N(z)) : \Im(F_N(z)) = 0 \} \right) \leq \sup_{|a_j| \leq 2^N} \rho_N(a_2, \dots, a_N).$$

Елементарним, але важливим геометричним спостереженням, яке виправдовує появу функції $\rho_N(a_2, \dots, a_N)$, є те, що вона напівнеперервна в горішній частині, отже, досягає максимального значення

$$\rho_N^{(0)} = \max_{|a_j| \leq 2^N} \rho_N(a_2, \dots, a_N).$$

З леми 2.8 випливає, що

$$J_N \leq \rho_N^{(0)} < 0. \quad (2.17)$$

Назвемо пару тригонометричних поліномів $\{C^{(0)}(t), S^{(0)}(t)\}$ за яких досягається максимум, *екстремальною парою*.

Переходимо до основної леми 2.9.

Поліном $f(z)$ як правило, є дійсним, якщо він дійсний в інтервалі $(-1, 1)$ і

в усіх інших точках диска $\mathbb{D}$ маємо $\Im\{f(z)\}\Im\{z\} > 0$. Позначимо множину усіх типово дійсних поліномів ступеню $\leq N$ як $\mathcal{T}_N$.

Лема 2.9. *Чинним є наступний обрахунок*

$$\rho_N^{(0)} \leq \sup_{F_N \in \mathcal{T}_N} \{F_N(-1)\}. \quad (2.18)$$

Наступну лему можна довести шляхом виснажливих, але простих обчислень [?, лема 6].

Лема 2.10. *Для будь-якого $k = 0, 1, \dots, N - 1$,*

$$\begin{aligned} 2 \sum_{j=1}^{N-k} \sin \frac{(j+1)\pi}{N+2} \sin \frac{(j+k)\pi}{N+2} &= (N-k-1) \sin \frac{k\pi}{N+2} \sin \frac{\pi}{N+2} \\ &+ \frac{1}{2} \frac{\cos \frac{\pi}{N+2}}{\sin \frac{\pi}{N+2}} \left((N-k+3) \sin \frac{(k+1)\pi}{N+2} - (N-k+1) \sin \frac{(k-1)\pi}{N+2} \right). \end{aligned}$$

Теорема 2.1. *Маємо*

$$\sup_{F_N \in \mathcal{T}_N} \{F_N(-1)\} = -\frac{1}{4} \sec^2 \frac{\pi}{N+2} \quad (2.19)$$

а оптимальним поліномом є

$$F_N^{(0)}(z) = \frac{1}{U'_N \left(\cos \frac{\pi}{N+2} \right)} \sum_{j=1}^N U'_{N-j+1} \left(\cos \frac{\pi}{N+2} \right) U_{j-1} \left(\cos \frac{\pi}{N+2} \right) z^j.$$

Наслідок 2.1.1. *Маємо*

$$J_N = -\frac{1}{4} \sec^2 \frac{\pi}{N+2},$$

а оптимальним поліномом є

$$a_j^{(0)} = \frac{1}{U'_N \left(\cos \frac{\pi}{N+2} \right)} U'_{N-j+1} \left(\cos \frac{\pi}{N+2} \right) U_{j-1} \left(\cos \frac{\pi}{N+2} \right).$$

Доведення. З комбінації (2.17), (2.18) and (2.19) випливає, що

$$J_N \leq F_N^{(0)}(-1) = -\frac{1}{4} \sec^2 \frac{\pi}{N+2}.$$

Зважаючи, що

$$J_N \geq \inf_{z \in \mathbb{D}} \left\{ \Re \left(F_N^{(0)}(z) \right) : \Im \left(F_N^{(0)}(z) \right) = 0 \right\} = F_N^{(0)}(-1),$$

наслідок доведено.

Як наслідок теореми 2.1 отримаємо оцінку радіуса Кебе [20].

У [31] радіус R_N полінома Кебе було визначено як супремум радіусів дисків з центром у початку координат, накриваних $F_N(\mathbb{D})$, де $F_N(z)$ є довільним однолистним поліномом. Було показано, що $R_N \geq \frac{1}{4} \sec^2 \frac{\pi}{N+3}$ що складає покращення порівняно зі значенням $\frac{1}{4}$ заданим функцією Кебе $K(z) = \frac{z}{(1-z)^2}$ у відомій теоремі Кебе про одну чверть. Наведену нижче оцінку можна доповнити вищенаведеним обчисленням, використовуючи теорему 2.1 та однолистність поліномів (2.6).

Наслідок 2.1.2. Радіус Кебе R_N для сімейства однолистних поліномів ступеню N задовольняє

$$\frac{1}{4} \sec^2 \frac{\pi}{N+3} \leq R_N \leq \frac{1}{4} \sec^2 \frac{\pi}{N+2}.$$

Задача 2. Знайдемо рішення проблеми пошукової оптимізації J_N в класі однолистных многочленів [17]. Припустимо, T_N означає множину всіх типово дій-

сних поліномів $F_N(z) = z + \sum_{j=2}^N \alpha_j z^j$ ступеню N нормалізованого за $F_N(0) = 0$, $F'_N = 1$. Поставимо завдання знайти

$$J_N = \max_{F_N(z) \in T_N} \max_t |F_N(e^{it})|. \quad (2.20)$$

Очевидно, $|F_N(e^{it})| \leq \sum_{j=1}^N |\alpha_j|$, тоді

$$J_N \leq \max_{\alpha_1=1} \left\{ \sum_{j=1}^N |\alpha_j| : \sum_{j=1}^N \alpha_j \sin jt \geq 0, t \in [0, \pi] \right\},$$

з рівністю якщо $\alpha_j \geq 0$ для всіх j .

У [71] було припущено, що

$$J_N \leq \frac{1}{4} \csc^2 \frac{\pi}{2(N+2)}. \quad (2.21)$$

Це було доведено у [79], та було показано, що для N непарного, (2.21) перетворюється на рівність. Та більш того, був показаний екстремальний поліном:

$$\begin{aligned} \frac{z}{z^2 + 1 - 2z \cos \frac{\pi}{N+2}} - \frac{4 \cos^2 \frac{\pi}{2(N+2)}}{N+2} (1 + z^{N+2}) \frac{1-z}{1+z} \\ \times \left(\frac{z}{z^2 + 1 - 2z \cos \frac{\pi}{N+2}} \right)^2, \end{aligned} \quad (2.22)$$

однак його єдиність встановлена.

Доведено, що екстремальний поліном є єдиним (з точністю до знаків парних членів) для всіх N , а для парних N маємо

$$J_N = \frac{1}{4\nu^2},$$

де ν – найменший невід’ємний корінь $U'_{N+1}(x)$, де $U_j(x) = U_j(\cos t) = \frac{\sin(j+1)t}{\sin t} = 2^j x^j + \dots$ є поліномом Чебишева другого роду. Також наведено алгоритм знаходження коефіцієнтів крайнього многочлена та явну формулу для коефіцієнтів при N непарних.

Зауважмо, що для N непарного, $\sin \frac{\pi}{2(N+2)}$ найменшим невід’ємним коренем з $U_{N+1}(x)$.

Лема 2.11. *Маємо*

$$\det(4x^2(I + A) - (I - B)) = \frac{1}{2^{N+2}x} U_{N+1}(x) U'_{N+1}(x).$$

Лема 2.12. *Характеристичними числами пучка матриць $I + A - \lambda(I - B)$ є*

$$\left\{ \frac{1}{4\mu_j^2} \right\}_{j=1}^{\lfloor (N+1)/2 \rfloor}, \quad \left\{ \frac{1}{4\nu_j^2} \right\}_{j=1}^{N - \lfloor (N+1)/2 \rfloor}, \quad \text{де } \mu_j = \cos \frac{j\pi}{N+2}, \quad U'_{N+1}(\nu_j) = 0;$$

їх можна впорядкувати як

$$\frac{1}{4\mu_1^2} < \frac{1}{4\nu_1^2} < \dots < \frac{1}{4\nu_{(N-1)/2}^2} < \frac{1}{4\mu_{(N+1)/2}^2}$$

якщо N непарне та

$$\frac{1}{4\mu_1^2} < \frac{1}{4\nu_1^2} < \dots < \frac{1}{4\mu_{N/2}^2} < \frac{1}{4\nu_{N/2}^2}$$

якщо N парне.

Тепер знаходимо власні вектори пучка $I + A - \lambda(I - B)$.

Лема 2.13. *Головні власні вектори $Z_j^{(0)}$ пучка матриць $I + A - \lambda(I - B)$, відповідні власним значенням $\lambda_j = \frac{1}{4} \sec^2 \frac{\pi j}{N+2}$, задаються $Z_j^{(0)} = Z^{(0)}(\cos \frac{\pi j}{N+2})$,*

$j = 1, \dots, \lfloor (N+1)/2 \rfloor$, де

$$Z^{(0)} = \{(-1)^k U_{k-1}(x) U_k(x)\}_{k=1}^N.$$

Лема 2.14. Нехай ν_j невід'ємний корінь поліному $U'_{N+1}(x)$. Головні вектори $Z_j^{(1)}$, що відповідають характеристичним числам $\lambda_j = \frac{1}{4\nu_j^2}$ пучка матриць $I + A - \lambda(I - B)$ задано $Z_j^{(1)} = Z^{(1)}(\nu_j)$, $j = 1, \dots, \lfloor N/2 \rfloor$, де

$$Z^{(1)}(x) = \left\{ (-1)^{k-1} (U_{k-1}(x) U_k(x) - U_{N-k}(x) U_{N-k+1}(x) + \frac{N+1-2k}{N+1} U_N(x) U_{N+1}(x)) \right\}_{k=1}^N. \quad (2.23)$$

Наслідок 2.1.3. Головні вектори, відповідні характеристичним числам $\lambda_j = \frac{1}{4} \sec^2 \frac{\pi j}{N+2}$ пучка матриць $I + A - \lambda(I - B)$ можуть бути записані у формі $Z_j^{(1)} = Z^{(1)}(\cos \frac{\pi j}{N+2})$, $j = 1, \dots, \lfloor (N+1)/2 \rfloor$, де вектор $Z^{(1)}(x)$ задано через (2.23).

Лема 2.15. Для всіх $k = 1, \dots, N$,

$$2 \sum_{j=1}^{N-k} U_j(x) U_{j+k-1}(x) = \frac{1}{1-x^2} ((N-k) T_{k-1}(x) - T_{N+2}(x) U_{N-k-1}(x)),$$

де $T_j(x) = T_j(\cos t) = \cos jt = 2^{j-1} x^j + \dots$ є поліномом Чебишева першого порядку.

Наслідок 2.1.4. Нехай $\mu = \cos \frac{\pi(N+1)}{2(N+2)}$. Для всіх $k = 1, \dots, N$,

$$2 \sum_{j=1}^{N-k} U_j(\mu) U_{j+k-1}(\mu) = \frac{1}{1-\mu^2} ((N-k) (\mu U_{k-2}(\mu) - U_{k-3}(\mu)) + U_{k+1}(\mu)).$$

Тепер локалізуємо найменший невід'ємний корінь $U'_{N+1}(x)$.

Лема 2.16. Припустимо N є парним, а ν є найменшим невід'ємним коренем

$U'_{N+1}(x)$. Тоді $\nu \in (\cos \tau_1, \cos \tau_2)$, где $\tau_1 = \frac{(N+1)\pi}{2(N+2)}$ та $\tau_2 = \frac{N\pi}{2(N+1)}$.

Доведення. Поліном $U_{N+1}(x)$ не змінює своєї опуклості на проміжку $(0, \cos \frac{N\pi}{2(N+2)})$, зокрема на $(\cos \tau_1, \cos \tau_2)$. З цього витікає, що, $U'_{N+1}(x)$ є монотонним на розглядуваному проміжку. Знаки $U'_{N+1}(x)$ на кінцях такого інтервалу збігаються зі знаками функції $\Phi(t) = (N+3)\sin(N+1)t - (N+1)\sin(N+3)t$ на кінцях (τ_2, τ_1) . Обчислюючи $\Phi(\tau_1) = (-1)^{N/2}2\sin\frac{\pi}{2(N+2)}$ та $\Phi(\tau_2) = -(-1)^{N/2}(N+1)\sin\frac{\pi}{N+1}$, тому знаки $U'_{N+1}(\cos \tau_1)$ і $U'_{N+1}(\cos \tau_2)$ різні, що і потрібно було довести.

Тепер опишемо деякі властивості координат головного вектора (2.23).

Лема 2.17. Нехай для парного N , ν – найменший невід'ємний корінь з $U'_{N+1}(x)$, при множині

$$z_k(x) = (-1)^{k-1} \left(U_{k-1}(x)U_k(x) - U_{N-k}(x)U_{N-k+1}(x) + \frac{N+1-2k}{N+1}U_N(x)U_{N+1}(x) \right), \quad k = 1, \dots, N.$$

Тоді

$$(a) \quad z_k(\nu) = z_{N-k+1}(\nu), \quad k = 1, \dots, N/2;$$

$$(б) \quad z_{k+1}(\nu) > z_k(\nu) > 0, \quad k = 1, \dots, N/2 - 1.$$

Разом з (2.20) розглянемо допоміжну задачу: знайти

$$\hat{J}_N = \max_{a_1=1} \left\{ \sum_{j=1}^N \alpha_j : \sum_{j=1}^N \alpha_j \sin jt \geq 0, \quad t \in [0, \pi] \right\}. \quad (2.24)$$

Теорема 2.2. *Маємо*

$$\hat{J}_N = \begin{cases} \frac{1}{4} \csc^2 \frac{\pi}{2(N+2)} & \text{якщо } N \text{ непарне,} \\ \frac{1}{4} \csc^2 \vartheta & \text{якщо } N \text{ парне,} \end{cases}$$

де ϑ - найменший невід'ємний корінь рівняння

$$(N+3) \cos(N+1)\vartheta + (N+1) \cos(N+3)\vartheta = 0.$$

Знайдемо коефіцієнти екстремальних поліномів (2.24).

Теорема 2.3. *Нехай $\mu = \sin \frac{\pi}{2(N+2)}$ буде найменшим невід'ємним коренем $U_{N+1}(x)$, а ν - найменшим невід'ємним коренем $U'_{N+1}(x)$. Припустимо, що*

$$z_k^{(1)}(\mu) = (-1)^{k-1} \left(U_{k-1}(\mu)U_k(\mu) - U_{N-k}(\mu)U_{N-k+1}(\mu) + \frac{N+1-2k}{N+1} U_N(\mu)U_{N+1}(\mu) \right), \quad (2.25)$$

$$z_k^{(1)}(\nu) = (-1)^{k-1} \left(U_{k-1}(\nu)U_k(\nu) - U_{N-k}(\nu)U_{N-k+1}(\nu) + \frac{N+1-2k}{N+1} U_N(\nu)U_{N+1}(\nu) \right) \quad (2.26)$$

для $k = 1, \dots, N$ - число координат головних векторів пучків матриць $I + A - \frac{1}{4\mu^2}(I - B)$, $I + A - \frac{1}{4\nu^2}(I - B)$ відповідно. Тоді коефіцієнти екстремального по-

лінома задачі (2.24) задаються як

$$\begin{aligned}\alpha_k &= \frac{\sum_{j=1}^{N-k+1} z_j^{(s)} z_{j+k-1}^{(s)} - \sum_{j=1}^{N-k-1} z_j^{(s)} z_{j+k+1}^{(s)}}{\sum_{j=1}^N z_j^{(s)} z_j^{(s)} - \sum_{j=1}^{N-2} z_j^{(s)} z_{j+2}^{(s)}}, \quad k = 1, \dots, N-2, \\ \alpha_{N-1} &= \frac{z_1^{(s)} z_{N-1}^{(s)} + z_2^{(s)} z_N^{(s)}}{\sum_{j=1}^N z_j^{(s)} z_j^{(s)} - \sum_{j=1}^{N-2} z_j^{(s)} z_{j+2}^{(s)}}, \\ \alpha_N &= \frac{z_1^{(s)} z_N^{(s)}}{\sum_{j=1}^N z_j^{(s)} z_j^{(s)} - \sum_{j=1}^{N-2} z_j^{(s)} z_{j+2}^{(s)}}.\end{aligned}\tag{2.27}$$

Наслідок 2.3.1. Для кожного $N = 1, 2, \dots$ екстремальний поліном задачі (2.24) є унікальним.

Доведення. Для простого узагальненого власного значення відповідний підпростір головних векторів є одновимірним.

Для N непарних формули (2.25) і (2.27) можна спростити.

Теорема 2.4. Нехай N є непарним. Тоді коефіцієнти екстремального полінома будуть заданими як

$$\alpha_k = \frac{(-1)^{k-1}}{U'_N\left(\sin \frac{\pi}{2(N+2)}\right)} U'_{N-k+1}\left(\sin \frac{\pi}{2(N+2)}\right) U_{k-1}\left(\sin \frac{\pi}{2(N+2)}\right), \quad k = 1, \dots, N. \tag{2.28}$$

Зауважимо, що співвідношення $U_{k-1}(\mu)U_k(\mu) = (-1)^{k-1} |U_{k-1}(\mu)U_k(\mu)|$ передбачає, що

$$\begin{aligned}\alpha_k &= \frac{2\gamma^2\mu}{1-\mu^2} ((N-k+3) |U_{k-1}(\mu)U_k(\mu)| + (N-k+1) |U_{k-2}(\mu)U_{k-1}(\mu)|) \\ &> 0, \quad k = 1, \dots, N.\end{aligned}$$

Покажемо, що N парних коефіцієнтів екстремального полінома у задачі (2.24) є невід'ємними.

Теорема 2.5. *Нехай N є парним. Тоді коефіцієнти екстремального полінома задачі (2.24) невід'ємні.*

Розв'язання основної задачі. Коефіцієнти екстремальних поліномів задачі (2.24) невід'ємні [64], тож ці поліноми також є екстремальними для (2.20). Крім того, згідно з Наслідком 2.3.1 екстремальний поліном (2.25) є єдиним.

Отже, (2.20) має точно два екстремальні поліноми, пов'язані між собою $F_N^{(2)} = -F_N^{(1)}(-z)$, де $F_N^{(1)}(z)$ – екстремальний поліном (2.24) коефіцієнти якого можна визначити за (2.25) – (2.27).

Наведемо приклади.

Побудуємо кілька екстремальних поліномів і знайдемо за ними образи одиничного кола. Розглянемо $N = 6$ і $N = 7$. Для парних N знаходимо коефіцієнти за (2.26), (2.27), а для N непарних за (2.25), (2.27) і (2.28):

$$F_6(z) = z + \sum_{j=2}^6 \alpha_j z^j,$$

де $\alpha_2 = 1.36252\dots$, $\alpha_3 = 1.55595\dots$, $\alpha_4 = 1.22943\dots$, $\alpha_5 = 0.84332\dots$, $\alpha_6 = 0.37361\dots$; та

$$F_7(z) = z + \sum_{j=2}^7 b_j z^j,$$

де $b_2 = 1.44834\dots$, $b_3 = 1.77398\dots$, $b_4 = 1.55232\dots$, $b_5 = 1.32706\dots$, $b_6 = 0.75810\dots$, $b_7 = 0.43104\dots$. Обчислюємо $\nu = 0.19818\dots$, $\mu = 0.17364\dots$, $J_6 = F_6(1) = \frac{1}{4\nu^2} = 6.36485\dots$, $J_7 = F_7(1) = \frac{1}{4\mu^2} = 8.29085\dots$. Зображення верхнього одиничного півкола під картами $F_6(z)$ and $F_7(z)$ показано на рисунках 2.1 і 2.2.

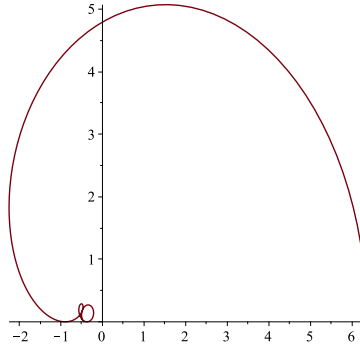


Рисунок 2.1: Зображення верхнього одиничного півкола під $F_6(z)$

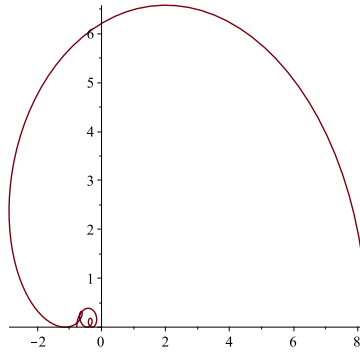


Рисунок 2.2: Зображення верхнього одиничного півкола під $F_7(z)$

Зауваження:

Припустимо $C(t) = \sum_{j=1}^N a_j \cos jt$ та $S(t) = \sum_{j=1}^N a_j \sin jt$ за умови, що всі a_j дійсні числа. У [24, 28] вирішено наступну задачу:

$$\sup_{a_1 + \dots + a_N = 1} \min_t \{C(t) : S(t) = 0\} = -\tan^2 \frac{\pi}{2(N+1)}. \quad (2.29)$$

Її рішення привело до нової процедури у проблемі стабілізації хаосу (див. [24, 28]). Це також можна інтерпретувати як задачу геометричного комплексного аналізу: найбільший інтервал виду $(-\mu, 1)$ який може бути охоплений оберненим зображенням доповнення до кола одиничного радіусу $(\mathbb{C} \setminus F_N(\mathbb{D}))^*$, де $z^* = 1/\bar{z}$ та $F_N(z) = a_1 z + \dots + a_N z^N$ при $F_N(1) = 1$, є $(-\cot^2 \frac{\pi}{2(N+1)}, 1)$. Як виявилось, екстремальними поліномами є відомі поліноми Саффріджа [91].

Нижче розглянемо подібну проблему, однак з іншою нормалізацією. Як наслідок, отримано теореми про покриття проміжків поліноміальними зображеннями $\mathbb{D}$. Ця робота продовжує дослідження, розпочате в [23].

Виходимо з припущення, що $a_1 = 1$. Нехай

$$J_N = \sup_{a_2, \dots, a_N} \min_t \{C(t) : S(t) = 0\}. \quad (2.30)$$

Доведемо наведену нижче теорему.

Теорема 2.6. *Маємо*

$$J_N = -\frac{1}{4} \sec^2 \frac{\pi}{N+2}.$$

Екстремальні поліноми є унікальними з коефіцієнтами, заданими формулами

$$a_j^{(0)} = \frac{1}{U'_N(\cos \frac{\pi}{N+2})} U'_{N-j+1} \left(\cos \frac{\pi}{N+2} \right) U_{j-1} \left(\cos \frac{\pi}{N+2} \right), \quad j = 1, \dots, N, \quad (2.31)$$

де $U_j(x)$ є поліномами Чебишева другого роду, а $U'_j(x)$ позначає похідну від $U_j(x)$.

Доведення теореми 2.6 містить дві частини. Перша полягає в приведенні:

$$J_N = \sup_{a_2, \dots, a_N} \{C(\pi) : S(t) \geq 0, 0 \leq t \leq \pi\}. \quad (2.32)$$

Ця частина є нестандартною і досить трудомісткою. Друга частина – обчислення (2.32). Вона також вимагає часу на обчислення, але ідеологічно наслідує схемі, запропонованій Л. Фейєром [24, 60]. А саме, використовуючи подання невід’ємних тригонометричних поліномів за Фейєром-Ріссом (2.32) до узагальненої задачі мінімізації частки Реле (див. [90, Розділ 6I], [66], [69]), можемо вирішити узагальнене характеристичне рівняння і отримати J_N як найменший корінь рівняння. Наступним кроком є отримання коефіцієнтів екстремального полінома з узагальне-

ного характеристичного вектора, що відповідає цьому узагальненому власному значенню. Незважаючи на чітку ідею, це все ще складне завдання. Як зазначено в [79], “Однак, загалом, нелегко адаптувати метод (...) для отримання однозначних результатів, зокрема, коли значення N велике”.

Для аргументації користуємось послідовністю лем. Повний виклад доведення можна знайти в [32].

Лема 2.18. *Існує функція, що обертає екстремуми до J_N .*

Задаємо $F_N(z) = z + a_2 z^2 + \dots + a_N z^N$. Тоді

$$J_N = \sup_{a_2, \dots, a_N} \min_t \{ \Re(F_N(e^{it})) : \Im(F_N(e^{it})) = 0 \}.$$

Наступна лема складає ядро першої частини доведення. Вона призводить до згортання (2.32).

Лема 2.19. *Якщо тригонометричний поліном $\Im(F_N(e^{it}))$ зникає в деякій точці $(0, \pi)$, то*

$$\min_t \{ \Re(F_N(e^{it})) : \Im(F_N(e^{it})) = 0 \} < J_N.$$

Позначимо

$$\begin{aligned} J_N^1 &= \sup_{a_2, \dots, a_N} \min_t \{ \Re(F_N(e^{it})) : \Im(F_N(e^{it})) > 0, 0 < t < \pi \} \\ &= \sup_{a_2, \dots, a_N} \{ F_N(-1) : \Im(F_N(e^{it})) > 0, 0 < t < \pi \}. \end{aligned}$$

З леми 2.19 випливає, що $J_N = J_N^1$. Далі, нехай

$$J_N^2 = \sup_{a_2, \dots, a_N} \{ F_N(-1) : \Im(F_N(e^{it})) \geq 0 \}. \quad (2.33)$$

Лема 2.20. Маємо $J_N^1 = J_N^2$.

Наступна лема є центральною у доведенні основного результату.

Лема 2.21. Маємо

$$J_N^2 = \max_{d_j} \left\{ -\frac{1 - d^T A d}{1 - d^T B d} : d^T d = 1 \right\} = -\min_{d_j} \left\{ \frac{d^T (I - A) d}{d^T (I - B) d} \right\}.$$

Зрозуміло, що матриці $I - A$ та $I - B$ позитивно визначені. Таким чином, проблема пошуку J_N зводиться до задачі на узагальнені власні значення [90, Зауваження 3, ст. 342]). А саме, нехай $\lambda_1 \leq \dots \leq \lambda_N$ – є коренями рівняння $\det((I - A) - \lambda(I - B)) = 0$. Тоді $J_N = -\lambda_1$.

За позитивною визначеністю $I - A$ та $I - B$ маємо $\lambda_1 > 0$. Відповідний мінімум досягається при узагальненому власному векторі $\delta^{(0)}$, який визначається із співвідношення $(I - A)\delta^{(0)} = \lambda_1(I - B)\delta^{(0)}$.

Наступна лема може бути цікава сама по собі. Це найскладніше твердження у другій частині доказу. Аргумент у цілком нетривіальний.

Лема 2.22.

$$\det(4x^2(I - A) - (I - B)) = \frac{1}{2^{N+2}x} U_{N+1}(x) U'_{N+1}(x).$$

Наслідок 2.6.1. Коренями рівняння $\det(4x^2(I - A) - (I - B)) = 0$ є

$$\{\pm\mu_j\}_{j=1}^{\lfloor (N+1)/2 \rfloor}, \quad \{\pm\nu_j\}_{j=1}^{N-\lfloor (N+1)/2 \rfloor},$$

де

$$\mu_j = \cos \frac{j\pi}{N+2}, \quad U'_{N+1}(\nu_j) = 0,$$

й їх можна впорядкувати в наступний спосіб:

$$0 < \mu_{(N+1)/2} < \nu_{(N-1)/2} < \cdots < \nu_1 < \mu_1 \quad \text{якщо } N \text{ є непарним,}$$

$$0 < \nu_{N/2} < \mu_{N/2} < \cdots < \nu_1 < \mu_1 \quad \text{якщо } N \text{ є парним числом.}$$

Лема 2.23. Нехай A і B – матриці, як у лемі 2.21. Розв’язання системи лінійних рівнянь

$$\left(4 \cos^2 \frac{j\pi}{N+2} (I - A) - (I - B) \right) \delta = 0, \quad j = 1, \dots, \left\lfloor \frac{N+1}{2} \right\rfloor,$$

є однопараметричним сімейством

$$c\delta^{(0)} \left(\cos \frac{j\pi}{N+2} \right),$$

де $c \in \mathbb{R}$ і

$$\delta^{(0)}(x) = (U_0(x)U_1(x), \dots, U_{N-1}(x)U_N(x))^T.$$

Наслідок 2.6.2.

$$\begin{aligned} J_N^2 &= -\lambda_1 = -\frac{1}{4\mu_1^2} = -\frac{1}{4} \sec^2 \frac{\pi}{N+2}, \\ \delta^{(0)} &= \left(U_0 \left(\cos \frac{\pi}{N+2} \right) U_1 \left(\cos \frac{\pi}{N+2} \right), \dots, \right. \\ &\quad \left. U_{N-1} \left(\cos \frac{\pi}{N+2} \right) U_N \left(\cos \frac{\pi}{N+2} \right) \right)^T. \end{aligned}$$

Таким чином, знайдені коефіцієнти d_j . Тепер маємо повернутися до α_j , а потім до a_j . Досить вдало реалізували спосіб запису для a_k 's. Наступна лема містить деякі корисні тригонометричні тотожності 2.24.

Лема 2.24. Для всіх $k = 0, 1, \dots, N - 1$,

$$2 \sum_{j=1}^{N-k} \sin \frac{(j+1)\pi}{N+2} \sin \frac{(j+k)\pi}{N+2} = (N-k-1) \sin \frac{k\pi}{N+2} \sin \frac{\pi}{N+2} + \frac{1}{2} \frac{\cos \frac{\pi}{N+2}}{\sin \frac{\pi}{N+2}} \left((N-k+3) \sin \frac{(k+1)\pi}{N+2} - (N-k+1) \sin \frac{(k-1)\pi}{N+2} \right).$$

З леми 2.24 отримуємо екстремальні коефіцієнти (2.31). Оскільки $J_N = J_N^1$, доведена теорема 2.6.

А саме, $|a_2| \leq 2 \cos 2\psi_N$, де $\psi_N = \pi/(N+3)$, якщо N непарний, та ψ_N - найменший позитивний корінь рівняння $(N+4) \sin(N+2)\psi_N + (N+2) \sin(N+4)\psi_N = 0$ якщо N парний. Можемо перевірити, що в останньому випадку $\pi/(N+3) < \psi_N < \pi/(N+2)$. Тоді передбачає оцінка за радіусом полінома Кебі:

$$R_N \geq \frac{1}{2 + \max |a_2|} \geq \frac{1}{4} \sec^2 \frac{\pi}{N+3}. \quad (2.34)$$

Наступна теорема оцінює зверху найбільший інтервал дійсної осі, який може бути покритий зображенням поліноміальної карти.

Теорема 2.7. Простий зв'язаний простір, що містить $F_N(\mathbb{D})$ для будь-якого відображення поліномів $F_N(z) = z + \sum_{j=2}^N a_j z^j$ містить інтервал

$$\left(-\frac{1}{4} \sec^2 \frac{\pi}{N+2}, \frac{1}{4} \sec^2 \frac{\pi}{N+2} \right). \quad (2.35)$$

Якщо F_N є типово дійсним, то інтервал (2.35) покривається $F_N(\mathbb{D})$.

Як показує приклад, не буде інтервалу більшого, ніж (2.35)

$$F_N(z) = z + \sum_{j=2}^N (-1)^{j-1} a_j^{(0)} z^j.$$

Наслідок 2.7.1. Радіус Кебе R_n^{tr} для типово дійсних поліномів задовольняє

$$R_N^{\text{tr}} \leq \frac{1}{4} \sec^2 \frac{\pi}{N+2}. \quad (2.36)$$

Зазначимо, що в [21, 91] для апроксимації функції Кебе використовувались поліноми Саффріджа. Кілька екстремальних властивостей поліномів Саффріджа було встановлено в [21, 24]. Зокрема, справедливим є наступне.

Теорема 2.8. *Мінімальна просто пов'язана множина, яка містить $F(\mathbb{D})$ для будь-якого поліноміального відображення $F_N(z) = \sum_{j=1}^N a_j z^j$ при $F_N(1) = 1$, містить інтервал*

$$\left(-\tan^2 \frac{\pi}{2(N+1)}, 1 \right). \quad (2.37)$$

Якщо F_N зазвичай дійсне, то інтервал (2.37) покривається $F_N(\mathbb{D})$.

Зауважимо, що різна нормалізація у задачах (2.29) та (2.30) призводить не тільки до різного значення екстремума, але й до різних екстремальних поліномів, які до того же виявляються новими.

Далі порівнюємо нашу задачу з задачею точних констант для поліноміального радіусу Бора. А саме, у 1914 році Гарольд Бор [6] винайшов таке

Теорема (Н. Bohr). *Припустимо, що ступеневий ряд $\sum a_k z^k$ сходиться в $\mathbb{D}$, а $|\sum a_k z^k| < 1$. Тоді $\sum |a_k z^k| < 1$ коли $|z| < 1/3$. Більш того, радіус $1/3$ є найкращим з можливих.*

У 2004 році Гуадаррама [46] ввів радіус Бора для поліномів. У 2007 році Рішар Фурньє [39] виявив, що точне значення радіусу є найменшим коренем визначника конкретної матриці Тауберіанівського типу. Обчислення цього значення все ще залишається відкритою проблемою. Зазначимо, що в нашому випадку

основна проблема була подібною, однак наша матриця була 5-діагональною, що дозволило нам знайти визначник.

Гарольд Боас та Дмитрій Хавінсон [5] виявили, що багатовимірний радіус Бора зникає із збільшенням розмірів зі швидкістю, зворотною до кореня виміру. Це було досить неочікуваним. Вони сформулювали проблему пошуку точного значення для фіксованого розміру, досі відкрито. Асимптотику у 2015 році знайшов Ченг Чу [15].

Задача 3. Перейдемо до формулювання задачі. Перед цим зробимо преамбулу.

Центральними об'єктами теорії однолистних поліномів є поліноми Саффріджа. У посібниках з однолистних функцій [10, 35, 38, 57, 60, 62, 63] ці поліноми складають основну частину розділу про однолисті поліноми. Поліноми, введені Т. Саффріджем [91] у 1969, можна явно записати виразом

$$S_{N,k}(z) = \sum_{j=1}^N \sigma_{j,k} z^j, \quad \sigma_{j,k} = \left(1 - \frac{j-1}{N}\right) \frac{\sin k\pi j/(N+1)}{\sin k\pi/(N+1)}, \quad k = 1, \dots, N.$$

Нескладно помітити, що $\sigma_{1,k} = 1$, та $\sigma_{N,k} = (-1)^{k-1}/N$.

Саффрідж відзначив деякі важливі властивості цих поліномів. Зокрема, він довів, що вони є однолистими (тобто однолистовими, $S_{N,k}(0) = 0$, і $S'_{N,k}(0) = 1$) на колі одиничного радіусу $\mathbb{D} = \{z : |z| < 1\}$.

Поліноми Саффріджа $S_{N,1}$ екстремальні в тому сенсі, що вони максимізують абсолютне значення коефіцієнтів будь-якого однолистного поліному порядку N з старшим коефіцієнтом $1/N$. Крім того, нулі їх похідних лежать на колі одиничного радіусу, а зображення такого одиничного кола містить $n - 1$ параболічних точок [16]. Таким чином, поліноми Саффріджа є квазіекстремальними поліномами у розумінні за Рушевейгом [40].

Компактна форма поліномів Саффіріджа була знайдена зовсім недавно [19]

$$S_{N,k}(z) = z \frac{N - 2(N+1)z \cos \frac{k\pi}{N+1} + (N+2)z^2 + (-1)^k(1-z^2)z^{N+1}}{N(z^2 + 1 - 2z \cos \frac{k\pi}{N+1})^2} \quad (2.38)$$

Це формулу можна отримати з теореми 1 [8], використовуючи для непарних k ([8], формулу (3)), і для парних ([8], формулу (4)), визначивши у цих формулах $n = N+1$, $j = \lfloor \frac{k+1}{2} \rfloor$ та $b_j = \frac{1}{2\sqrt{N}}$, $b'_j = b_j$ (k – непарне), $c_j = \frac{1}{2\sqrt{N}}$, $c'_j = \gamma_j$ (k – парне).

Звідки витікає, що поліном $S_{N,1}(z)$ буде визначатись через $S_N(z)$.

Подання за (2.38) надає нам змогу обчислити

$$S_N(-1) = -\frac{1}{4} \frac{N+1}{N} \sec^2 \frac{\pi}{2(N+1)},$$

й, таким чином,

$$\lim_{N \rightarrow \infty} S_N(-1) = -\frac{1}{4}. \quad (2.39)$$

У [8] екстремальну задачу оцінки максимального значення модулю однолистових поліномів порядку N на одиничному колі $\mathbb{D}$ було розв'язано. Показано, що екстремальний поліном є $S_N(z)$, а максимальне значення

$$S_N(1) = \frac{1}{4} \frac{N+1}{N} \csc^2 \frac{\pi}{2(N+1)}.$$

У [28] наведено нові екстремальні властивості поліномів $S_{N,k}(z)$. Зокрема, поліном Саффіріджа $S_N(z)$, нормований так, що $S_N(1) = 1$ (тобто $\frac{1}{S_N(1)}S_N(z)$), є єдиним оптимізатором для задачі

$$\sup_{\sum_{j=1}^N a_j = 1} \left(\inf_{z \in \mathbb{D}} \{ \operatorname{Re}(F_N(z)) : \operatorname{Im}(F_N(z)) = 0 \} \right),$$

де $F_N(z) = \sum_{j=1}^N a_j z^j$ є довільним (не обов'язково однолистним) поліномом порядку N .

Розв'язання цієї екстремальної задачі, а також нетандартні застосування екстремальних поліномів для розв'язання задач управління в нелінійних дискретних системах наведено в [24, 33, 56].

Екстремальні властивості поліномів Саффіріджа подібні до екстремальних властивостей відомої функції Кебе. Це вказує на можливість розглядати поліноми Саффіріджа як природній замінник цієї функції. У [20] поліном $\frac{-1}{4S_N(-1)}S_N(z)$ було використано для апроксимації функції Кебе у сенсі підпорядкування Андрієвського-Рущевейга [1], тобто поліном Саффіріджа $S_N(z)$ нормалізовано так, що точка -1 переходить до $-1/4$. Важливість цієї проблеми обумовлена тим, що функція Кебе є екстремальною у двох відомих теоремах геометричного комплексного аналізу: у теоремі П. Кебе про чверть [58] та в гіпотезі Л. Бібербаха [4] – тепер відомій як теорема Л. де Бранжа [96].

Аналогічні екстремальні задачі для T -симетричної однолистової функції $f^{(T)}(z) = [f(z^T)]^{1/T}$ набагато складніші. Видається, точні нерівності, за цих умов недосяжні навіть для $T = 2$. Однак, можна спробувати хоча б зрозуміти асимптотичну поведінку.

Спочатку розглянемо T -симетричну функцію Кебе

$$K^{(T)}(z) = \frac{z}{(1 - z^T)^{2/T}}.$$

Більш того, як зазначено вище, якщо $f(z)$ – однолистова функція, то $f^{(T)}(z) = [f(z^T)]^{1/T}$ є T -симетричною однолистною функцією. Якщо $p(z)$ – однолистовий поліном, то $[p(z^T)]^{1/T}$ не може бути поліномом. Це можливо лише тоді, коли по-

ліном $p(z)$ має представлення $p(z) = z[q(z)]^T$, де $q(z)$ – поліном за нормалізації $q(0) = 1$.

Досліджуючи поліноми Саффіріджа, легко спостерегти, що функція $[S_N(z^T)]^{1/T}$ не є поліномом. В якості заміни розглянемо новий клас T - симетричних поліномів $F_N^{(T)}(z) = z(a_1 + a_2 z^T + \dots + a_N z^{(N-1)T})$ з нормалізацією $F_N^{(T)}(1) = 1$. Первинно вони були введені в [33] в контексті знаходження нестійких циклів для дискретних динамічних систем. Зауважимо, що $F_N^{(1)}(z) = \frac{1}{S_N(1)} S_N(z)$, та крім того, наступна асимптотична оцінка еквівалентна (12) у [33]:

$$F_N^{(T)}(e^{i\pi/T}) \sim c_T N^{-2/T}. \quad (2.40)$$

Тут $A_N \sim B_N$ означає, що $\lim_{N \rightarrow \infty} \frac{A_N}{B_N} = 1$, та $c_T = \pi^{\frac{2}{T}-1} \Gamma^2(\frac{1}{T} + \frac{1}{2})$, при $\Gamma(\zeta)$ звичайній гамма-функції. Досліджуючи ренормалізовану функцію $\left[\frac{S_N(z^T)}{S_N(1)} \right]^{1/T}$, бачимо, що існує подібна асимптотична оцінка

$$\left[\frac{S_N(-1)}{S_N(1)} \right]^{1/T} \sim 2^{-2/T} \pi^{2/T} N^{-2/T}$$

що відповідає (2.40) для $T = 1$.

Задача 4. Довести, що для $N \rightarrow \infty$, $\max \left| \frac{1}{a_1} F_N^{(T)}(z) \right| \sim \frac{1}{2^{2/T} c_T} N^{2/T}$. Варто зазначити, що це значення виявляється кращим, ніж отримане для функції $[S_N(z^T)]^{1/T}$. Другим основним результатом є теорема 2.12, яка показує, що поліноми $\frac{1}{a_1} F_N^{(T)}(z)$ з належним наближенням репрезентують T - симетричну функцію Кебе.

Наступна теорема є ключовою для знаходження асимптотичної оцінки максимального модулю T - симетричних однолистових поліномів в $\mathbb{D}$. Вважаємо, що ця теорема може також представляти інтерес й незалежно від цього аспекту.

Теорема 2.9. Для довільного дійсного x маємо

$$\lim_{N \rightarrow \infty} \prod_{j=1}^N 2 \left(1 - \cos \pi \frac{2j-1}{2N+x} \right) = 2^{1-x}, \quad (2.41)$$

$$\lim_{N \rightarrow \infty} \frac{1}{N} \prod_{j=1}^N 2 \left(1 - \cos \pi \frac{2j}{2N+1+x} \right) = 2^{1-x}. \quad (2.42)$$

Віповідно до [33] побудуємо T - симетричні поліноми $F_N^{(T)}(z) = z(a_1 + a_2 z^T + \dots + a_N z^{(N-1)T})$ як інтерполяційні поліноми Лагранжа. Для непарних N вибираємо вузли

$$e^{i\psi_1}, e^{i\psi_2}, \dots, e^{i\psi_{(N-1)/2}}, \quad \psi_k \in (0, \pi),$$

і будуємо породжуючий поліном

$$\eta_N(z) = z \prod_{j=1}^{\frac{N-1}{2}} (z - e^{i\psi_j})(z - e^{-i\psi_j}).$$

У разі парного N вибираємо вузли

$$e^{i\psi_1}, e^{i\psi_2}, \dots, e^{i\psi_{(N-2)/2}}, -1, \quad \psi_k \in (0, \pi),$$

та породжуючий поліном

$$\eta_N(z) = z(z+1) \prod_{j=1}^{\frac{N-2}{2}} (z - e^{i\psi_j})(z - e^{-i\psi_j}).$$

Обираємо $\psi_k = \pi \frac{2+T(2k-1)}{2+T(N-1)}, k = 1, \dots, \lfloor \frac{N-1}{2} \rfloor$.

Породжуючий поліном може бути представлений у стандартній формі

$$\eta_N(z) = \sum_{j=1}^N \gamma_j z^j.$$

Коефіцієнти можна визначити за формулами Вієта; зокрема, очевидно, що $\gamma_1 = \gamma_N = 1$. Використовуючи породжуючий поліном $\eta_N(z)$, визначаємо поліноми

$$q_N(z) = \eta_N(z) \left(\frac{A_0}{z} + 2A_1 \sum_{k=1}^{\frac{N-1}{2}} \frac{z - \cos \psi_k}{(z - e^{i\psi_k})(z - e^{-i\psi_k})} \right),$$

якщо N непарне, або

$$q_N(z) = \eta_N(z) \left(\frac{A_0}{z} + \frac{A_1}{z+1} + 2A_1 \sum_{k=1}^{\frac{N-2}{2}} \frac{z - \cos \psi_k}{(z - e^{i\psi_k})(z - e^{-i\psi_k})} \right),$$

якщо N парне, де

$$A_0 = \frac{2}{\eta_N(1)} \frac{1 + T(N-1)}{2 + T(N-1)}, \quad A_1 = -\frac{2}{\eta_N(1)} \frac{T}{2 + T(N-1)}.$$

Тоді $q_N = \sum_{j=1}^N a_j z^{j-1}$, де $a_1 = A_0$ й відшукуваний T - симетричний поліном може бути представлений як

$$F_N^{(T)}(z) = z q_N(z^T),$$

й дотримано вимоги нормалізації $F_N^{(T)}(1) = 1$.

У [33] висунули припущення, що поліноми $F_N^{(T)}(z)$ є однолистними; ця здогадка підтверджується численними доказами. Якщо припущення вважати істин-

ним, поліном

$$S_N^{(T)}(z) = \frac{1}{a_1} F_N^{(T)}(z)$$

буде T - симетричним однолистовим поліномом. При $T = 1$ маємо $S_N^{(1)}(z) = S_N(z)$, які точно є поліномами Саффріджа.

Записуємо

$$S_N^{(T)} = \sum_{j=1}^N B_j z^{1+T(j-1)}, \quad B_k = \frac{a_k}{a_1}.$$

Обчислимо

$$B_j = \left(1 - \frac{(j-1)T}{1 + (N-1)T}\right) \gamma_j \quad (2.43)$$

(див [33]). Отже, $B_1 = 1$, і $B_N = \frac{1}{1+(N-1)T}$. Зауважимо, що B_N має максимально можливе значення для будь-якого однолистного поліному ступеня $1 + (N-1)T$, що задовольняє $B_1 = 1$.

Отримаємо основний результат з оцінки максимального модуля $S_N^{(T)}(z)$ на $\mathbb{D}$. Для цього знадобиться кілька допоміжних формул та формулювань.

По-перше, звернемо увагу, що для N непарних,

$$q_N(-1) = \eta_N(-1) \left(-A_0 - \frac{N-1}{2} A_1 \right) = -\frac{\eta_N(-1)}{\eta_N(1)} = \frac{\prod_{j=1}^{\frac{N-1}{2}} (1 + \cos \psi_j)}{\prod_{j=1}^{\frac{N-1}{2}} (1 - \cos \psi_j)}$$

якщо навіть N парне,

$$q_N(-1) = A_1 \lim_{z \rightarrow -1} \frac{\eta_N(z)}{z+1} = -\frac{2T}{2+T(N-1)} \frac{1}{\eta_N(1)} \lim_{z \rightarrow -1} \frac{\eta_N(z)}{z+1} =$$

$$\frac{T}{2+T(N-1)} \frac{\prod_{j=1}^{\frac{N-2}{2}} (1 + \cos \psi_j)}{\prod_{j=1}^{\frac{N-2}{2}} (1 - \cos \psi_j)}.$$

Більш того, асимптотична оцінка за (2.40) дає

$$q_N(-1) \sim c_T N^{-2/T} \text{ для } N \rightarrow \infty. \quad (2.44)$$

Далі знаходимо асимптотичну оцінку коефіцієнту a_1 .

Лема 2.25. $\lim_{N \rightarrow \infty} \frac{|q_N(-1)|}{a_1} = 2^{-2/T}$.

Покажемо, що коефіцієнти поліному $S_N^{(T)}(z)$ є невід'ємними.

Лема 2.26. $B_j > 0$ для всіх $1 \leq j \leq N$.

Перейдемо до доведення основного результату.

Теорема 2.10. При $N \rightarrow \infty$ виконується наступна вимога асимптотичної оцінки:

$$\max_{z \in \mathbb{D}} \left| S_N^{(T)}(z) \right| \sim \frac{1}{2^{2/T} c_T} N^{2/T}, \quad (2.45)$$

$$\text{де } c_T = \pi^{\frac{2}{T}-1} \Gamma^2\left(\frac{1}{T} + \frac{1}{2}\right).$$

Порівняймо асимптотичну оцінку, отриману в теоремі (2.10) з подібною оцінкою для T -симетричної функції $[S_N(z^T)]^{1/T}$. Відповідно до теореми Бранд-

та [8, 9]

$$|S_N(1)|^{1/T} = \left[\frac{1}{4} \frac{N+1}{N} \sec^2 \frac{\pi}{2(N+1)} \right]^{1/T} \sim \pi^{-2/T} N^{2/T}.$$

Для дослідження асимптотики відношення $\frac{|S_N(1)|^{1/T}}{|S_N^{(T)}(1)|}$, визначаємо

$$k(T) = \frac{2^{2/T} c_T}{\pi^{2/T}} = \pi^{-1} 2^{\frac{2}{T}} \Gamma^2\left(\frac{1}{T} + \frac{1}{2}\right).$$

Лема 2.27. Для всіх $T = 1, 2, \dots$ маємо

$$k(T) \leq 1. \quad (2.46)$$

Таким чином, асимптотична оцінка максимального модулю на колі одиничного радіусу $\mathbb{D}$ для полінома $S_N^{(T)}(z)$ краща (більший діапазон); тоді для функції $[S_N(z^T)]^{1/T}$ для всіх T . Чисельні обчислення показують, що діапазон модулю полінома $S_N^{(T)}(z)$ більший не тільки асимптотично, а й для всіх N ; наприклад,

$$S_3^{(5)}(1) = 1.37 \dots > [S_3(1)]^{1/5} = 1.17 \dots$$

Розглянемо проблему апроксимації T – симетричної функції Кебе. В [1] доведена відома теорема про апроксимацію однолистных функцій на $\mathbb{D}$ однолистними поліномами.

Теорема 2.11. Існує константа $c > 0$, така що для кожної однолистной функції $f(z)$ на $\mathbb{D}$, існує послідовність однолистных поліномів $p_N(z)$ у $\mathbb{D}$, з нормалізацією $p_N(0) = f(0)$, при чому

$$f(\rho_N \mathbb{D}) \subset p_N(\mathbb{D}) \subset f(\mathbb{D}), \quad \rho_N = 1 - c/N,$$

для всіх $N > 2c$.

Крім того, згідно [44, 45], $\pi \leq c \leq 73$, порядок величини $1 - \rho_N$ має бути $1/N$ при $N \rightarrow \infty$.

Д. Дімітров [20] показав, що нормалізовані поліноми Саффріджа є хорошим наближенням функції Кебе. Точніше, автор показав, що $(-\infty, -\frac{1}{4}) \not\subset \frac{-1}{4S_N(-1)}S_N(\mathbb{D})$, звідки $\frac{-1}{4S_N(-1)}S_N(\mathbb{D}) \subset K(\mathbb{D})$. Й більш того, існує таке ρ_N , при якому $K(\rho_N\mathbb{D}) \subset \frac{-1}{4S_N(-1)}S_N(\mathbb{D})$; ρ_N може бути оцінено з огляду на вищенаведене, $\rho_N \leq \rho$, де ρ має задовільняти рівнянню $K(\rho) = \frac{-1}{4S_N(-1)}S_N(1)$. Таким чином, $\frac{\rho}{(1-\rho)^2} = \frac{1}{4} \cot^2 \frac{\pi}{2N+2}$ й відповідно

$$\rho = \frac{1 - \sin \pi/(2N+2)}{1 + \sin \pi/(2N+2)} \sim 1 - \frac{\pi}{N}.$$

Більш того, можна продемонструвати, що $\rho_N = \rho$ [45].

Доведемо, що подібна верхня межа досягається, якщо здійснити апроксимацію T -симетричної функції Кебе $K^{(T)}(z) = \frac{z}{(1-z^T)^{2/T}}$ за допомогою T -симетричних поліномів Саффріджа. Вважаємо $S_N^{(T)}(z)$ однолистними, і нормалізуємо їх так, щоб $K^{(T)}(e^{i\pi/T}) = e^{i\pi/T} 2^{-2/T}$, тобто будемо розглядати нормалізований поліном $\frac{e^{i\pi/T} 2^{-2/T}}{S_N^{(T)}(e^{i\pi/T})} S_N^{(T)}(z)$.

Теорема 2.12. Нехай ρ_N найбільше число, так що виконуються

$$K^{(T)}(\rho_N\mathbb{D}) \subset \frac{e^{i\pi/T} 2^{-2/T}}{S_N^{(T)}(e^{i\pi/T})} S_N^{(T)}(\mathbb{D}).$$

Тоді $\rho_N \leq \rho$, де ρ задовольняє

$$\rho \sim \left(1 - \frac{2\sqrt{c_T}}{N}\right)^{1/T}, \quad c_T = \pi^{\frac{2}{T}-1} \Gamma^2\left(\frac{1}{T} + \frac{1}{2}\right). \quad (2.47)$$

Зауважимо, що у теоремі (2.12) нерівність не можна замінити знаком рівно-

сті. Тим не менше, чисельні експерименти показують, що оцінка радіусу близька до точної і покращується, коли N набуває більшого значення. Більш того, асимптотична формула (2.47) дуже проста у використанні, деякі приклади чого наведемо нижче.

Якщо $T = 1$, отримуємо $\rho = 1 - \frac{2\sqrt{c_1}}{N}$, а оскільки $c_1 = \pi^2/4$, маємо $\rho \sim 1 - \frac{\pi}{N}$, що збігається з оцінкою з [45]. На рисунку 2.3 показано зображення кола одиничного радіусу при відображенні $\frac{30 \cos^2 \pi/62}{31} S_{30}(z)$, а зображення кола радіусу $\rho = \frac{1 - \sin \pi/(2 \cdot 30 + 2)}{1 + \sin \pi/(2 \cdot 30 + 2)}$ при відображенні $K(z)$.

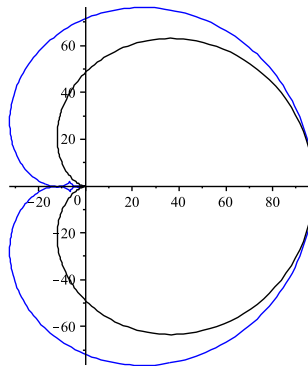


Рисунок 2.3: Зображення кола одиничного радіусу при відображенні $\frac{30 \cos^2(\pi/62)}{31} S_{30}(z)$ (синій) та кола радіусу $\rho = \frac{1 - \sin \pi/(2 \cdot 30 + 2)}{1 + \sin \pi/(2 \cdot 30 + 2)}$ при відображенні $K(z)$ (чорний).

При $T = 2$ отримаємо $\rho \sim (1 - \frac{2}{N})^{1/2} \sim 1 - \frac{1}{N}$.

При $T = 3$ обчислюємо $c_3 = 0.869\dots$, $(c_3)^{3/2} = 0.811\dots$ й, застосувавши (2.47) отримуємо $\rho \sim (1 - \frac{1.622}{N})^{1/3}$ що, після підстановки $N = 30$ дає нам для ρ наближене значення 0.981. При подальшому обчисленні отримуємо $\frac{e^{i\pi/3} 2^{-2/3}}{S_{30}^{(3)}(e^{i\pi/3})} = 0.989\dots$ На рисунку 2.4 показано зображення кола одиничного радіусу при відображенні $0.989 \cdot S_{30}^{(3)}(z)$ та кола радіусом $\rho = 0.981$ при відображенні $K^{(3)}(z)$. Таким чином, знову пересвідчуємося у дуже добрій апроксимації.

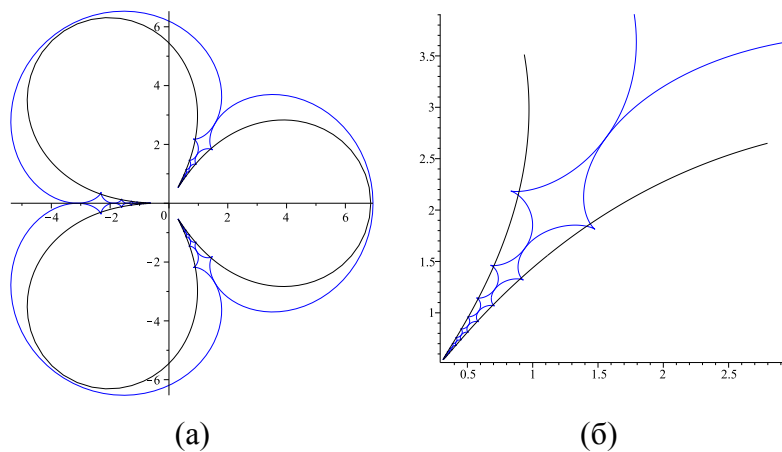


Рисунок 2.4: Зображення кола одиничного радіусу (а) (та масштабована версія (б)) при відображенні $0.989 \cdot S_{30}^{(3)}(z)$, порівняно з колом радіусу $\rho = 0.981$ при відображенні $K^{(3)}(z)$.

2.3 Предикативний контроль

Розглянемо нелінійну дискретну систему (1.3) з нерухомою точкою η . Передбачається, що ця система має одну чи кілька нестійких нерухомих точок, тобто $\eta = f(\eta)$. Оскільки нерухомі точки не відомі, то невідомий і спектр матриць Якобі. Для положення рівноваги $x_n = \eta$ системи [1.3] спектр матриці Якобі позначимо $\mu_1, \dots, \mu_m$.

Також розглянемо систему управління

$$x_{n+1} = F(x_n) \quad (2.48)$$

де $F(x) = \sum_{j=1}^N \vartheta_j f^{(j)}(x)$, $f^{(1)} = f(x)$, $f^{(1)}(x) = f(x)$, $f^{(k)} = f(f^{(k-1)}(x))$, $k = 2, \dots, N$. Числа $\vartheta_1, \dots, \vartheta_N$ дійсні. Неважко перевірити, що при $\sum_{j=1}^N \vartheta_j = 1$ у системі (2.48) також є положення рівноваги η . Завдання полягає у виборі параметра N і коефіцієнтів $\vartheta_1, \dots, \vartheta_N$ так, щоб становище рівноваги η системи (2.48) було локально асимптотично стійким. Природно, що при побудові цих коефіцієнтів необхідно використовувати інформацію щодо різних випадків локалізації мульт-

типлікаторів $\{\mu_1, \dots, \mu_m\} \subset M$.

Лема 2.28. Матрицю Якобі положення рівноваги η системи (2.48) можна подати у вигляді

$$\sum_{j=1}^N \vartheta_j J^j, \quad (2.49)$$

де J - матриця Якобі положення рівноваги η системи (1.3).

Замість системи (2.48) можна розглядати іншу систему управління

$$x_{n+1} = f \left(\vartheta_1 x_n + \sum_{j=2}^N \vartheta_j f^{(j-1)}(x_n) \right). \quad (2.50)$$

При $\sum_{j=1}^N \vartheta_j = 1$ у системі (2.50) положення рівноваги зберігається. Крім того, матриця Якобі цього положення рівноваги виражається через матрицю Якобі цього ж положення рівноваги системи (1.3) за формулою (2.49).

Перевага системи управління (2.50) перед системою (2.48) – це менша кількість обчислень функції $f(x)$. Всі результати для системи (2.48) переносяться без зміни на систему (2.50).

Теорема 2.13. Нехай $f \in C^1$ і у системи (1.3) є нестійке положення рівноваги з мультиплікаторами $\mu_1, \dots, \mu_m$. Тоді це положення буде локально асимптотично стійким положення рівноваги (2.48), якщо

$$r(\mu_j) \in D, j = 1, \dots, m,$$

де $D = \{z \in \mathbb{C} : |z| < 1\}$ – відкрите центральне одиничне коло, $r(\mu) = \sum_{j=1}^N \vartheta_j \mu^j$.

Відзначимо, що обов'язково має виконуватися умова $r(1) = 1$.

Різні оцінки на мультиплікатори дозволяють конструювати системи управління, що стабілізують положення рівноваги.

Випадок $M = \{\mu_1, \dots, \mu_m\}$

Якщо мультиплікатори відомі точно, то можна вибрати $N = m + 1$ і коефіцієнти $\vartheta_1, \dots, \vartheta_{m+1}$ з умови $r(\mu) = \sum_{j=1}^{m+1} \vartheta_j \mu^j = \frac{\mu}{\prod_{k=1}^m (1 - \mu_k)} \prod_{k=1}^m (\mu - \mu_k)$.

Якщо початкова точка належить басейну тяжіння положення рівноваги, то жбіжність до циклу надлінійна. Це випливає з того, що всі мультиплікатори положення рівноваги системи (2.48) виявляються рівними нулю.

Розглянемо основні можливі випадки локалізації мультиплікаторів нелінійної системи (2.48).

Випадок $M = (-\mu^*, \mu^*)$, $\mu^* > 1$.

З теореми (2.13) витікає, що для стабілізації положення рівноваги потрібно побудувати поліном $r(\mu)$, так щоб $r(1) = 1$ та $|r(\mu)| \leq 1$ для всіх $|\mu| < \mu^*$. Потрібний поліном можна побудувати, використовуючи поліноми Чебишева першого роду $T_N(x) = \cos(n \arccos x)$.

Теорема 2.14. Нехай $f \in C^1$ і у системи (1.3) нестійке положення рівноваги з мультиплікаторами у системи $\{\mu_1, \dots, \mu_m\} \subseteq (-\mu^*, \mu^*)$. Нехай величина N вибирається з умови $\frac{1}{\sin \pi/2N} > \mu^*$, а коефіцієнти $\vartheta_1, \dots, \vartheta_N$

$$r(\mu) = \sum_{j=1}^N \vartheta_j \mu^j = (-1)^{\frac{N-1}{2}} T_N \left(\mu \sin \frac{\pi}{2N} \right),$$

де $T_N(x)$ – поліном Чебишева першого роду непарного порядку N . Тоді це положення рівноваги буде локально асимптотично стійким положенням рівноваги системи (2.48) (виключаючи кінцеве число випадків, коли $\mu_j = \frac{\cos \pi k/N}{\sin \pi/2N}$, $k = 1, \dots, N-1$).

Випадок $M = (0, \mu^*)$, $\mu^* > 1$.

Теорема 2.15. Нехай $f \in C^1$ і у системи (1.3) є нестійке становище рівноваги з мультиплікаторами $\{\mu_1, \dots, \mu_m\} \subseteq (0, \mu^*)$. Нехай величина N вибирається з умови $\frac{1}{1+2\cos\frac{\pi}{2N}} \cot^2 \frac{\pi}{4N} > \mu^*$, а коефіцієнти $\vartheta_1, \dots, \vartheta_N$ з

$$r(\mu) = \sum_{j=1}^N \vartheta_j \mu^j = -T \left(\mu \left(\cos \frac{\pi}{N} - \cos \frac{\pi}{2N} \right) + \cos \frac{\pi}{2N} \right),$$

де $T_N(x)$ – поліном Чебишева першого роду порядку N . Тоді це положення рівноваги буде локально асимптотично стійким положенням рівноваги системи (2.48) (виключаючи кінцеве число випадків).

Випадок $M = (-\mu^*, 1)$, $\mu^* > 1$.

Теорема 2.16. Нехай $f \in C^1$ і у системи (1.3) є нестійке становище рівноваги з мультиплікаторами $\{\mu_1, \dots, \mu_m\} \subseteq (-\mu^*, 1)$. Нехай величина N вибирається з умови $\cot^2 \frac{\pi}{4N} > \mu^*$, а коефіцієнти $\vartheta_1, \dots, \vartheta_N$ з

$$r(\mu) = \sum_{j=1}^N \vartheta_j \mu^j = T_N \left(\mu \left(1 - \cos \frac{\pi}{2N} \right) + \cos \frac{\pi}{2N} \right),$$

де $T_N(x)$ – поліном Чебишева першого роду порядку N . Тоді це положення рівноваги буде локально асимптотично стійким положенням рівноваги системи (2.48) (виключаючи кінцеве число випадків).

Лема 2.29. Нехай $f \in C^1$ і у системи (1.3) є нестійке становище рівноваги з мультиплікаторами $\{\mu_1, \dots, \mu_m\} \subseteq (-\mu^*, 1)$. Нехай величина ν задовольняє умовам $\frac{-1+\mu^*}{1+\mu^*} < \nu < 1$. Тоді це положення рівноваги буде локально асимптотично стійким положенням рівноваги системи

$$x_{n+1} = (1 - \nu)f(x_n) + \nu x_n.$$

Лему 2.29 можна використовувати для вирішення завдання стабілізації у випадку 2.2. Функція $(-1)^{\frac{N-1}{2}} T_N \left(\mu \sin \frac{\pi}{2N} \right)$ при N непарно показує інтервал $(-\infty, \mu^*)$ в $(-c^*, 1)$. Функція $-T_N \left(\mu \left(\cos \frac{\pi}{N} - \cos \frac{\pi}{2N} + \cos \frac{\pi}{2N} \right) \right)$ при N непарно преобразує інтервал $(-\infty, \mu^*)$ в $(-c^*, 1)$, а при N парне $(-\infty, \infty)$ в $(-c^*, 1)$. Це означає, що мультиплікатори положення рівноваги системи (2.48) при виборі коефіцієнтів, як зазначено в Теоремі 2.15 або 2.16, лежать в інтервалі $(-c^*, 1)$. Таким чином, у системи

$$x_{n+1} = (1 - \nu)F(x_n) + \nu x_n,$$

де $F(x) = \sum_{j=1}^N \vartheta_j f^{(j)}(x)$ або $F(x) = f \left(\vartheta_1 x + \sum_{j=2}^N \vartheta_j f^{(j-1)}(x) \right)$, мультиплікатори положення рівноваги лежать в інтервалі $(-1, 1]$. Тобто положення рівноваги стає локально асимптотично стійким (за винятком, можливо, кінцевого числа випадків).

У частному випадку при $N = 2$, $-T_N \left(\mu \left(\cos \frac{\pi}{N} - \cos \frac{\pi}{2N} \right) + \cos \frac{\pi}{2N} \right) = 2x - x^2$, а при $N = 3$, $(-1)^{\frac{N-1}{2}} T_N \left(\mu \sin \frac{\pi}{2N} \right) = \frac{3}{2}x - \frac{1}{2}x^3$.

Використовуючи ідеї теореми (2.13) можна запропонувати наступну схему стабілізації положення рівноваги (при цьому коефіцієнти ϑ_j не будуть обов'язково константами).

1) знаходимо матрицю $f'(x)$,

2) знаходимо характеристичний поліном цієї матриці $\sum_{j=1}^{m+1} \vartheta_j(x) \mu^{j-1}$,

3) унормовуємо характеристичний поліном $\frac{1}{\sum_{j=1}^{m+1} \vartheta_j(x)} \sum_{j=1}^{m+1} \vartheta_j(x) \mu^{j-1}$,

4) будуємо систему управління $x_{n+1} = F(x_n)$,

де $F(x) = \frac{1}{\sum_{j=1}^{m+1} \vartheta_j(x)} \sum_{j=1}^{m+1} \vartheta_j(x) f^{(j)}(x)$ або

$$F(x) = f \left(\frac{1}{\sum_{j=1}^{m+1} \vartheta_j(x)} \left(\vartheta_1 x + \sum_{j=2}^{m+1} \vartheta_j(x) f^{(j-1)}(x) \right) \right).$$

2.4 Модифікаційні схеми градієнтного спуску

Розгляньмо завдання пошуку стаціонарних точок функції $z = \Phi(\xi_1, \dots, \xi_m)$, для чого необхідно вирішити систему рівнянь $grad\Phi = 0$, де $grad\Phi = \left(\frac{\partial\Phi}{\partial\xi_1}, \dots, \frac{\partial\Phi}{\partial\xi_m} \right)^T$ (символом «Т» позначено операцію транспонування). Надалі будемо позначати $grad\Phi = G(x)$, де $x = (\xi_1, \dots, \xi_m)^T$. Матриця Гессе $H = \left\{ \frac{\partial^2\Phi}{\partial\xi_i\partial\xi_j} \right\}_{i,j=1}^{m,m}$ симетрична, отже, її власні значення $\{\tau_1, \dots, \tau_m\}$ дійсні. Якщо ці власні значення вираховані у стаціонарній точці, різних знаків, то розглянута стаціонарна точка – сідлова, інакше екстремальна. Розглядаємо загальний випадок знаків власних значень.

Стаціонарна точка є нерухомою точкою відображення $x - \gamma G(x)$, де γ - деяке ненульове число. Для знаходження стаціонарної точки розглянемо наступну ітераційну схему

$$x_{n+1} = x_n - \gamma G(x_n). \quad (2.51)$$

Тобто, тепер можна розглядати завдання пошуку нерухомої точки системи (1.3), в якій $f(x) = x - \gamma G(x)$, і застосовувати всі схеми, запропоновані в попередньому розділі. Мультиплікатори нерухомої точки пов'язані з власними значеннями матриці Гессе за формулами

$$\mu_j = 1 - \gamma\tau_j, j = 1, \dots, m.$$

Якщо серед τ_j є числа різних знаків, то серед мультиплікаторів циклу обов'язково знайдуться значення більше одиниці.

Нехай $\tau_1 \leq \dots \leq \tau_m$, причому $\tau_1 < 0$, $\tau_m > 0$. Тоді при додатному γ отримуємо, що $\mu_1 = 1 - \gamma\tau_1 > 1$, $\mu_m = 1 - \gamma\tau_m < 1$. Нехай додатково $\mu_m < 0$, і $\mu^* = \max\{\mu_1, |\mu_m|\}$. Тоді $\mu_j \in \{-\mu^*, \mu^*\}$, і для побудови схеми управління можна застосувати Теорему 2. Виберемо $N = 3$ тоді схема керування (2.48) прийме вигляд $x_{n+1} = \frac{3}{2}f(x_n) - \frac{1}{2}f(f(f(x_n)))$, а (2.50) $x_{n+1} = f\left(\frac{3}{2}x_n - \frac{1}{2}f(f(x_n))\right)$. Запишемо ці схеми для системи (2.51):

Схема 1.1

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = u_n - \gamma G(u_n) \\ w_n = v_n - \gamma G(v_n) \\ x_{n+1} = \frac{3}{2}u_n - \frac{1}{2}w_n \end{cases}. \quad (2.52)$$

Схема 1.2

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = u_n - \gamma G(u_n) \\ w_{n+1} = \frac{3}{2}x_n - \frac{1}{2}v_n \\ x_{n+1} = w_n - \gamma G(w_n) \end{cases}. \quad (2.53)$$

Теорема 2.17. Нехай $G \in C^1$ і параметр γ вибирається з умови $\max\{1 - \gamma\tau_1, |1 - \gamma\tau_m|\} < 2$. Тоді нерухома точка $x_n = \eta$ різноманітних схем 2.52 и 2.53 буде локально стійким положенням рівноваги.

Застосовуймо тепер Теорему 2.15. Нехай $\tau_1 \leq \dots \leq \tau_m$, причому $\tau_1 < 0$, $\tau_m > 0$ и $\mu_m = 1 - \gamma\tau_m \geq 0$. Тоді $\mu_j \in [0, 1 - \gamma\tau_1]$, $j = 1, \dots, m$. Виберемо $N = 2$, тоді схема керування (2.48) прийме вигляд $x_{n+1} = 2f(x_n) - f(f(x_n))$, а (2.50) $x_{n+1} = f(2x_n - f(x_n))$. Запишемо ці схеми для системи (2.51):

Схема 2.1

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = u_n - \gamma G(u_n) \\ x_{n+1} = 2u_n - v_n \end{cases} \quad (2.54)$$

Схема 2.2

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = 2x_n - u_n \\ x_{n+1} = v_n - \gamma G(v_n) \end{cases} \quad (2.55)$$

Теорема 2.18. Нехай $G \in C^1$ і параметр γ задовольняє умови $\gamma < \frac{\sqrt{2}-1}{|\tau_1|}$. Тоді нерухома точка $x_n = \eta$ різностних схем 2.54 і 2.55 буде локально стійким положенням рівноваги.

Застосуємо додатково Лемму 2.29. Отримаємо схеми:

Схема 3.1

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = u_n - \gamma G(u_n) \\ x_{n+1} = (1 - \nu)(2u_n - v_n) + vx_n \end{cases} \quad (2.56)$$

Схема 3.2

$$\begin{cases} u_n = x_n - \gamma G(x_n) \\ v_n = 2x_n - u_n \\ x_{n+1} = (1 - \nu)(v_n - \gamma G(v_n)) + vx_n \end{cases} \quad (2.57)$$

Теорема 2.19. Нехай $G \in C^1$ і параметри γ і ν вибираються з умов $\gamma < \frac{1}{|\tau_1|}$, $\frac{2\mu-\mu^2-1}{2\mu-\mu^2+1} < \nu < 1$, де $\mu = 1 - \gamma\tau_m$. Тоді нерухома точка $x_n = \eta$ різних схем 2.56 і 2.57 буде локально стійким положенням рівноваги.

Зазначимо, що кожна наведена схема дозволяє знаходити відразу всі стаціонарні точки, параметри при цьому повинні вибиратися згідно з умовами Теорем.

Однак, басейни тяжіння можуть бути дуже різними при переході від однієї схеми до іншої. Для ефективності застосування різних схем важливі як знання про басейни тяжіння стаціонарних точок, так і швидкість збіжності до стаціонарної точки. Непараметрично це регулюється вибором схеми, параметрично - вибором кроку і додаткового параметра ν .

2.5 Висновки по другому розділу

У другому розділі був розроблений метод пошуку параметрів для запропонованих модифікацій методів градієнтного спуску з використанням зворотнього зв'язку із затриманням та прогнозуючого контролю для дискретних динамічних систем на основі вирішення трьох задач оптимізації в різних класах комплексних многочленів.

Для кожної задачі оптимізації були виявлені екстремайзери та їх коефіцієнти, які забезпечують їх виконання для широкого спектру завдань оптимізації.

Розроблено алгоритм пошуку оптимальних параметрів, що складається з трьох етапів: рішення оптимізаційної задачі на класі комплексних поліномів, проведення чисельного моделювання з використанням варіацій допустимих параметрів, експериментальна перевірка знайдених параметрів для різних завдань

РОЗДІЛ 3

МЕТОДИ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ МОДИФІКАЦІЙ МЕТОДУ ГРАДІЄНТНОГО СПУСКУ

3.1 Пошук екстремальних та сідлових точок

Розглянемо тестову мінімаксну задачу

$$\min_{x \in \mathbf{R}} \max_{y \in \mathbf{R}} \left[\Phi(x, y) = 2x^2 + y^2 + 4xy + \frac{4}{3}y^2 - \frac{1}{4}y^4 \right], \quad (3.1)$$

де $x, y \in \mathbf{R}$. Вирішуючи рівняння $grad\Phi = 0$ знаходимо всі точки екстремумів

$$z_0 = (0.0);$$

$$z_1 = (-2 + \sqrt{2}, 2 - \sqrt{2});$$

$$z_2 = (-2 - \sqrt{2}, 2 + \sqrt{2}).$$

Точка z_1 є точкою локального мінімуму, а z_0 и z_2 – сідлові точки.

Побудуємо басейни тяжіння екстремальних точок завдання (3.1) для кожної схеми з попереднього розділу.

Моделювання та візуалізація численних результатів виконані мовою програмування Python з дистрибутива Anaconda. Графіки побудовані за допомогою бібліотеки Matplotlib. Всі обчислення проводилися на комп'ютері з Intel[®] Xeon[®] E-2286M CPU @ 2.40GHz с 64GB ECC RAM з встановленою операційною системою Ubuntu Linux (64 bit).

У процесі обчислень не використовувалися ніякі спеціалізовані бібліотеки підвищеної точності, оскільки перевірялася стійкість методів до стандартних апаратних похибок. При цьому ітеративні схеми виконувалися тисячі разів на базі арифметики кінцевої точності, що призводило до округлень результатів. Виника-

ли ситуації, коли результатом обчислень було число NaN (Not a number), це означало, що результати наближалися до однієї з нескінченностей або виходили за межі прямокутника на площині (x, y) . Такі випадки оброблялися спеціальним чином і початкові точки не додавалися в басейни тяжінь. Сітка будувалася в інтервалах $x \in [-8.0, 6.0]$ и $y \in [-2.0, 6.0]$ з кроком 0.1. Точність бралася 10^{-6} . Для забезпечення умов порівняння різних схем параметри вибиралися однаковими: $\gamma = 0.1$ и $N = 5000$, у схемах 3.1 (2.57) і 3.2 (2.57) додатковий параметр $\nu = 0.5$.

У схемах 3.1 (2.56) і 3.2 (2.57) додатковий параметр ν . Ліловим кольором позначено басейн тяжіння точки z_0 , блакитним кольором - z_1 , жовтим - z_2 . З таблиці 3.1 видно значну відмінність басейнів тяжіння при застосуванні різних схем. Для проведеного експерименту обидва різновиди схеми 2 змогли зійтися з більшого числа початкових точок (64 – 64.6%). Відсоток сбіжності обчислювався як відсоток точок з сітки, яку використовували як початкові точки ітерації, для яких була схожість до стаціонарної точки.

Для розуміння оцінки сбіжності були вираховані стандартні статистичні дані по швидкості сбіжності до кожної зі стаціонарних точок схем з різних точок басейну тяжіння. Дані представлені в таблиці 3.2, всі наведені значення - це кількість ітерацій, витрачених для наближення до стаціонарних точок на задану відстань.

Всі запропоновані схеми використовували постійне значення параметра γ , що з практичної точки зору не є оптимальним, і багато бібліотечних функцій динамічно змінюють цей параметр у міру наближення до стаціонарної точки. Тому дані схеми повинні бути модифіковані з урахуванням зміни кроку в міру наближення до стаціонарної точки.

3.2 Тестові функції

Розглянемо одну з функцій, що часто використовується для оцінки продуктивності різних алгоритмів оптимізації. Вона була запропонована Ховардом Розенброком як функція для якої пошук глобального мінімуму є нетривіальним завданням. Запропонована модифікація методу градієнтного спуску (1.5) є новим оптимізаційним алгоритмом і тому має сенс порівняти його можливості у пошуку глобального мінімуму функції Розенброка. Також необхідно порівняти швидкості збіжності між модифікованим методом та основними модифікаціями градієнтного спуску із модуля оптимізації бібліотеки PyTorch. Даний пакет застосовується у навчанні нейронних мереж як у дослідних цілях так і при вирішенні індустріальних завдань.

Функція Розенброка для двох змінних має вигляд

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

Ця функція має глобальний мінімум у точці $(x, y) = (1, 1)$, де $f(x, y) = 0$. Графік цієї функції містить великий яр, усередині якого і має бути глобальний мінімум.

Саме з цієї причини градієнтні методи мають проблеми в пошуку, так як вони роблять велику кількість переміщень з одного боку яру на іншу, що й уповільнює досягнення екстремальної точки. Графік такого наближення зображено на наступному малюнку 3.2. При цьому, чим ближчий алгоритм до локального мінімуму, тим частіше він переміщується між стінками.

Перед тим, як порівняти продуктивність алгоритмів, з метою майбутнього використання нових розширень бібліотеки PyTorch при тренуванні нейронних

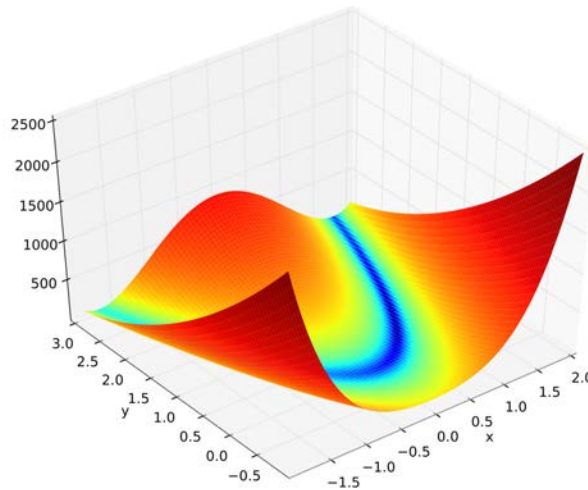


Рисунок 3.1: Яр Розенброка [99]

мереж, функція Розенброка була представлена у вигляді тензорних операцій, для яких можливе використання вбудованого в PyTorch алгоритму Autograd. Цей алгоритм обчислює приватні похідні функції по кожному з її параметрів, а саме по x і y .

Можливість такого уявлення дозволила розширити стандартний інтерфейс модуля *torch.optim* новим алгоритмом та налагодити його на даній простій функції. Це дозволило під час тестування використовувати весь модуль оптимізації із заміною лише одного рядка виклику оптимізатора.

Під час тесту було використано три алгоритми: Adam, SGD із включеним моментом та модифікаціями Нестерова ([73]) та алгоритмом (1.5). Для пошук мінімуму використовувалось 40 кроків. Результат порівняння подано на графіку 3.3.

З графіка 3.3 видно, наскільки краще працює контроль, який дозволив алгоритму (1.5) швидше досягти мінімуму функції і не здійснювати більшості відхилень від траєкторії, які зазнавали алгоритми Adam і SGD [92].

Окрім функції Розенброка були проведені чисельні експерименти на стандартному наборі тестових функцій, спеціально розроблених для порівняння різних оптимізаційних алгоритмів. Наведемо перелік цих функцій (таблиця 3.3).

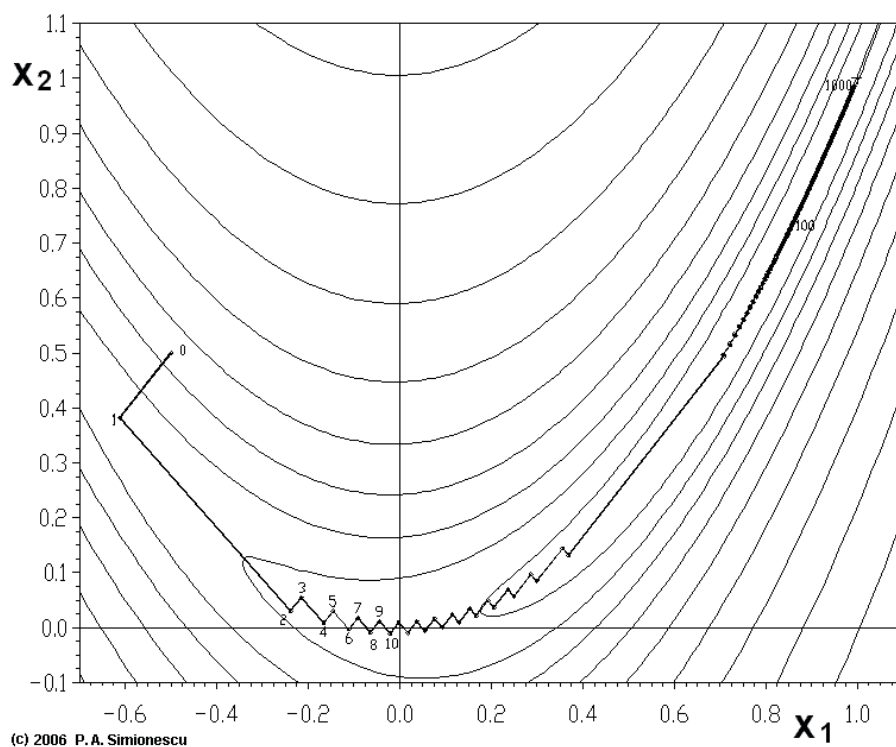
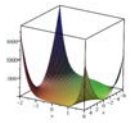
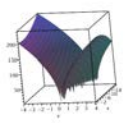
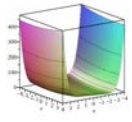
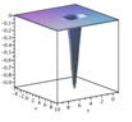
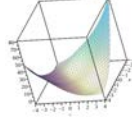
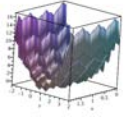
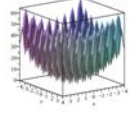
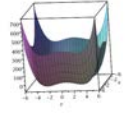


Рисунок 3.2: Траєкторія пошуку мінімуму яру [89]

Таблиця 3.3: Вигляд тестових функцій

Beale		Bukin	
Three-hump camel		Easom	
McCormik		Lewi	
Rastrigin		Styblinski–Tang	

Для кожної з цих функцій зроблені численні порівняння сбіжності до локальних мінімумів, та побудовані басейни тяжіння. Запропоновані схеми показали стабільну поведінку при вирішенні тестових завдань, змогли знайти більше локальних мінімумів з басейнами тяжіння більшої площі.

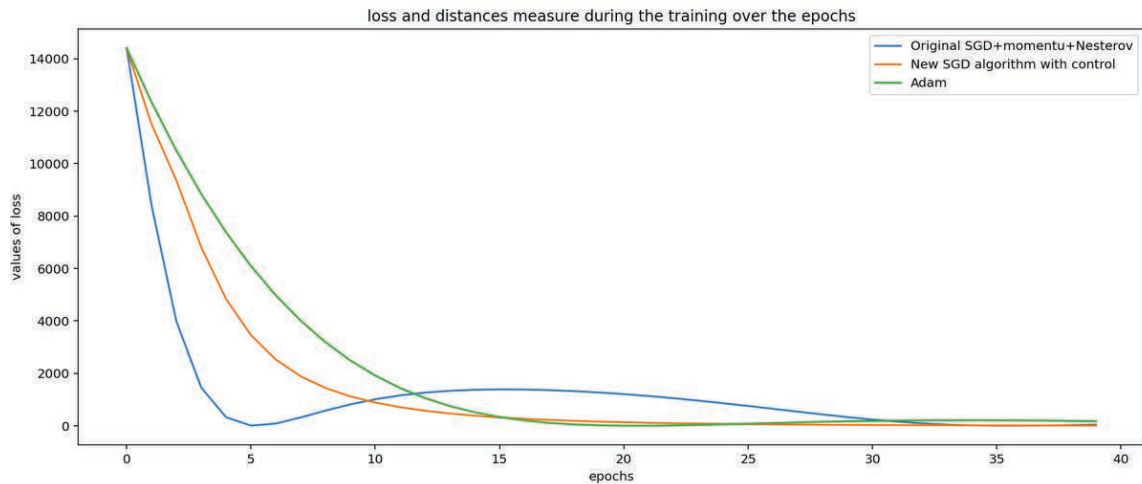


Рисунок 3.3: Результат порівняльного аналізу алгоритмів Adam, SGD+Nesterov та алгоритма (1.5)

3.3 Навчання багат шарового персептрона для розпізнавання рукописних цифр

У 2006 році почалася чергова хвиля популярності нейронних мереж, початком якого можна вважати статтю Хінтона ([50]). У цій статті було продемонстровано можливості глибоких нейронних мереж для розпізнавання рукописних цифр порівняно із Support Vector Machines (SVM) на наборі даних MNIST. Дана база містить шістдесят тисяч зображень, призначених для тренування і десять тисяч для перевірки результатів. Наведемо приклад зображень із цієї бази



Рисунок 3.4: Приклад зображень цифр із бази MNIST

Для проведення порівняльного експерименту на перевірку продуктивності нових модифікацій алгоритму градієнтного спуску, було реалізовано багаторівневий перцептрон наступної архітектури (рис. 3.5).

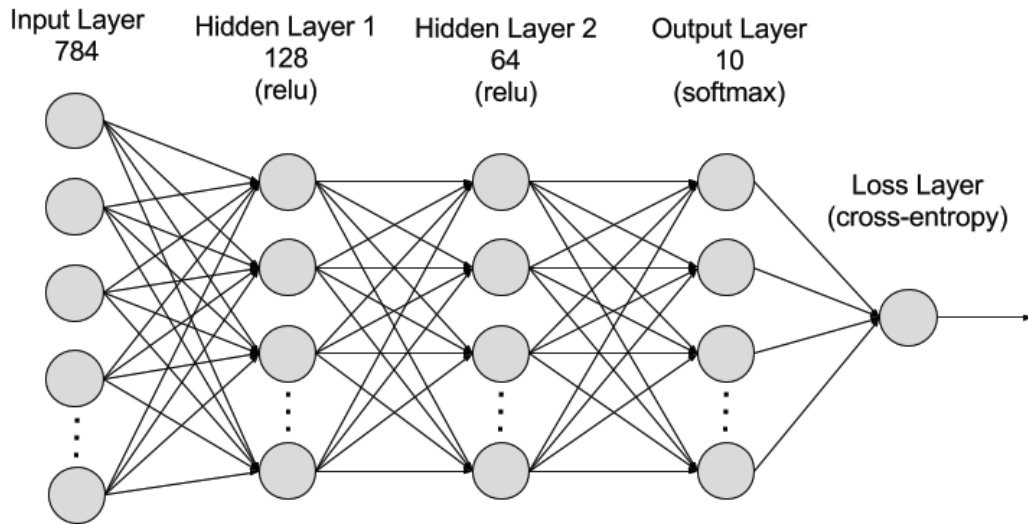


Рисунок 3.5: Багаторівневий перцептрон для розпізнавання рукописних цифр із бази MNIST

Вибір кількості скритих рівнів і нейронів на кожному з цих рівнів проводився на основі аналізу великої кількості численних експериментів з різними значеннями цих гіперпараметрів. При цьому, оптимізація була по глибині, так як велика глибина архітектури приводила до підвищення часу навчання, а також вимагала більшого обсягу навчальних даних.

Важливою особливістю всіх експериментів є те, що всі експерименти починаються з однієї і тієї ж точки багатомірного простору. Для цього проводився додатковий контроль поведінки генератора випадкових чисел (вибиралось одне і те ж передпочтакове насіння).

Після установки генератора випадкових чисел були ініціалізовані початкові значення для кожного параметра нейронної мережі. Ці параметри були збережені в спеціальному файлі і підгружалися в начале кожного тренування.

Ці процедури створили однакові початкові умови для експериментів, що до-

зволило коректно порівняти результати як за швидкостями сходження, так і порівняти отримані точки локальних мінімумів функцій втрат для різних оптимізаційних алгоритмів.

Експерименти показали, що перцептрон з двома скритими шарами дозволяє досягти високих результатів на тестовому наборі даних.

Проведено порівняння продуктивності нейронної мережі, швидкості процесу навчання та часу витраченого на досягнення кращого результату для кількох алгоритмів. Серед них був стандартний SGD (з моментом і модифікаціями Нестєрова) та нова модифікація градієнтного спуску за формулою (1.5).

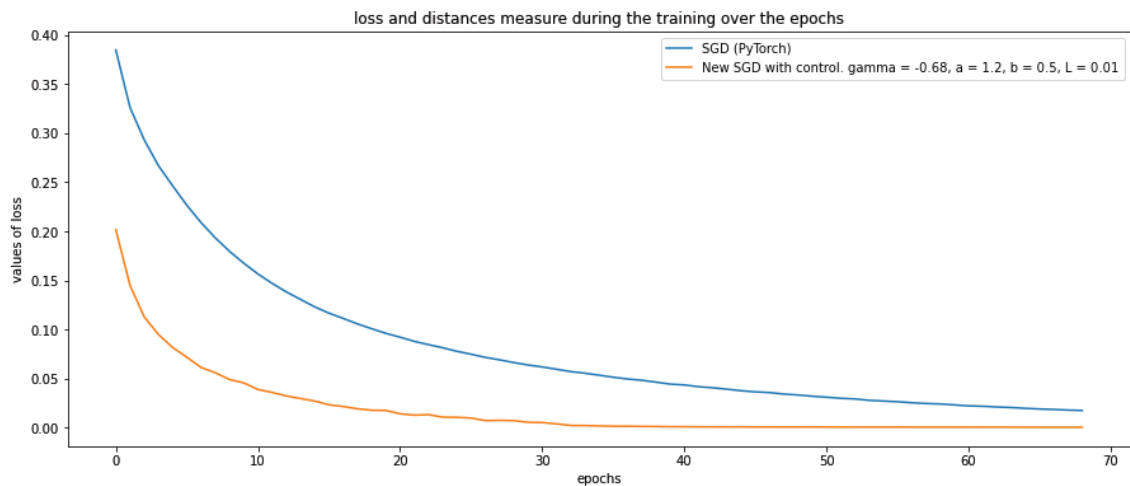


Рисунок 3.6: Порівняння SGD та алгоритма (1.5)

З графіка видно, що вдалося підібрати такі параметри a , b і γ , у яких алгоритм (1.5) зміг досягти меншого значення функції втрати за меншу кількість кроків. Відобразимо на графіку це співвідношення (рис. 3.7).

Модифікація методу (1.5) досягла значення меншого ніж 0.01734 за 21 епоху. Це значення було мінімальним для стандартного SGD, досягнутого ним на 68 епосі. Все це доводить практичні можливості нового алгоритму при складній повторності функції втрати та швидким наближенням до локального мінімуму.

Із малюнка (3.8) слід високу відповідність алгоритму (1.5) щодо SGD Так як

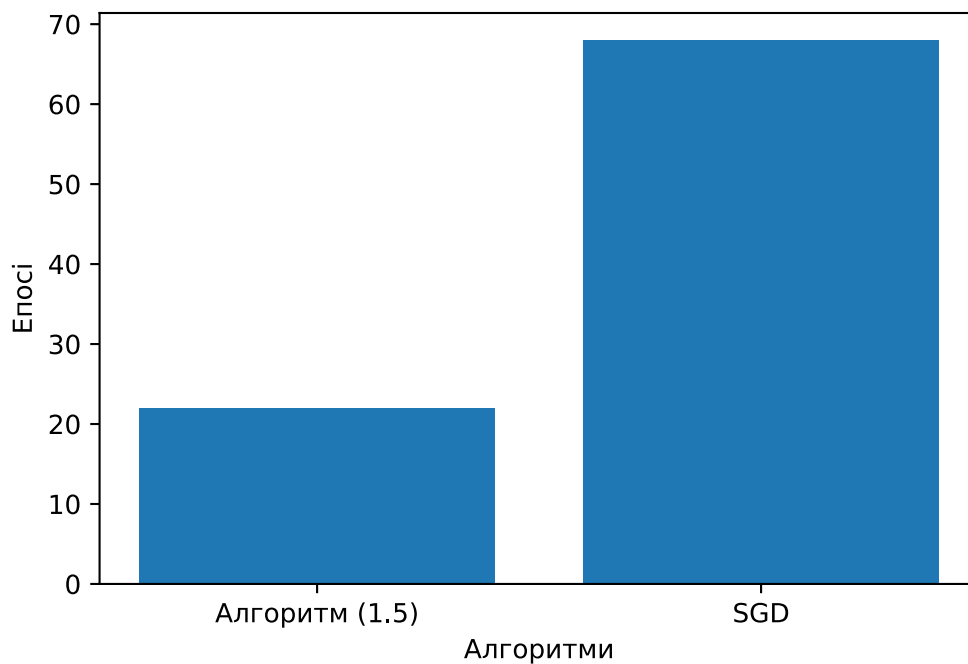


Рисунок 3.7: Порівняння швидкості досягнення локального мінімуму в процесі навчання нейронної мережі двома різними модифікаціями градієнтного спуску (графік згенеровано бібліотекою Matplotlib у Python)

перцептрон має більш високу розмірність, ніж розглянуті перед цими тестовими функціями, то отримані результати демонструють предсказанні теорією можливості нової модифікації градієнтного спуска.

Зображення (3.9) показує ймовірності, які повертають навчальний багаторівневий перцептрон, коли на вході подається цифра 3. Показано, що існує невелика ймовірність того, що це цифра 8 (ймовірність менше 0.2) і ймовірність більша 0.8 для цифри 3.

Реалізація модифікації методу градієнтного спуску із запізненням (1.5) в бібліотеці PyTorch показала 20% прискорення процесу навчання багаторівневого перцептрону щодо алгоритму SGD.

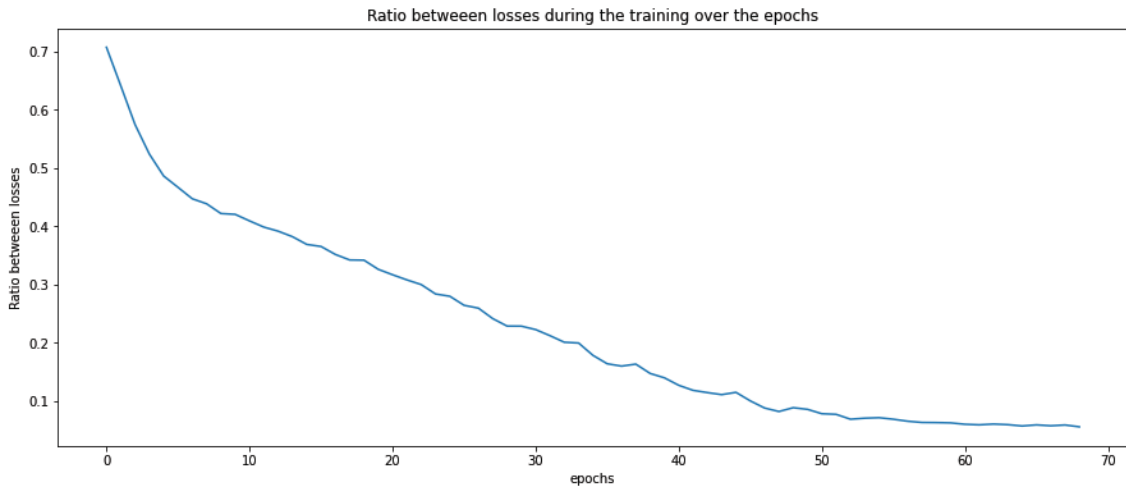


Рисунок 3.8: Відношення значень функцій втрат при навчанні градієнтних алгоритмів (1.5) та SGD

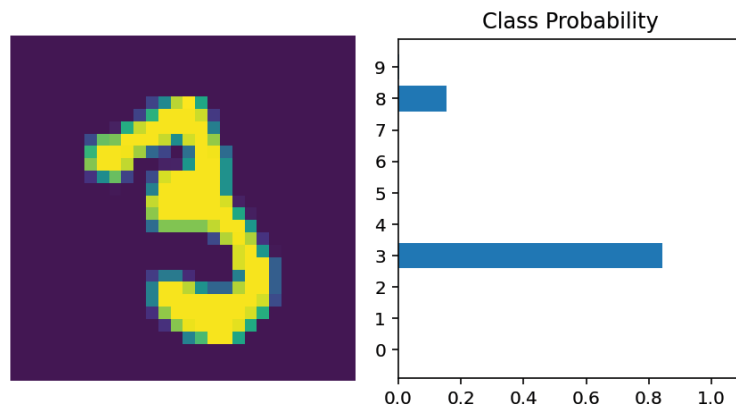


Рисунок 3.9: Визуалізація результатів роботи нейронної мережі для цифри 3

3.4 Висновки до третього розділу

У третьому розділі набули подальшого розвитку методи навчання нейронних мереж на основі запропонованих модифікацій методу градієнтного спуску. Проведено їх дослідження для різних архітектур нейронних мереж.

Розроблено на основі (1.5) методи навчання нейронних мереж на базі модифікації методу градієнтного спуску з урахуванням зворотнього зв'язку з запізненням в ДДС (1.5) при заданих параметрах $T = 1$, $N = 2$.

Набули подальшого розвитку метод навчання нейронних мереж на основі модифікації методу градієнтного спуску з урахуванням прогнозуючого зворотно-

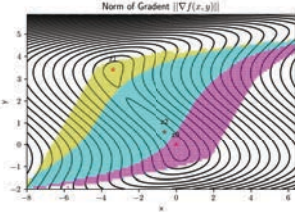
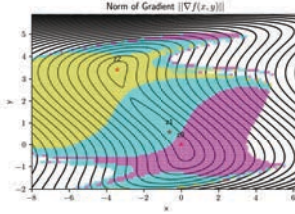
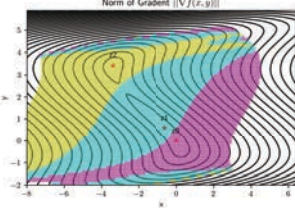
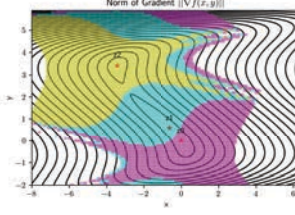
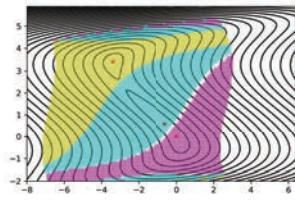
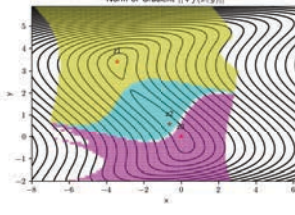
го зв'язку в ДДС (1.6) при заданих параметрах $T = 1$, $N = 2$.

Ці схеми було покладено в основу створення низки нових схем реалізації методу градієнтного спуску. Проведено їх тестування на прикладах тестових функцій, таких як функція Розенброка.

Проведено навчання багатошарового персептрона для розпізнавання рукописних цифр на базі бібліотеки PyTorch в якому нові модифікації показали 20% прискорення процесу навчання.

Таким чином, набули подальшого розвитку методи навчання нейронних мереж на основі запропонованих модифікацій методу градієнтного спуску, що дозволило зменшити час навчання нейронних мереж різних архітектур при вирішенні задач сегментації та класифікації об'єктів (**третій пункт новизни наукового дослідження**).

Таблиця 3.1: Басейни тяжіння стаціонарних точок для різноманітних схем 1.1 (2.52), 1.2 (2.53), 2.1 (2.54), 2.2 (2.55), 3.1 (2.56), 3.2 (2.57)

	<i>Схема 1.1</i>	<i>Схема 1.2</i>
		
площі басейнів	$z_0 - 1515$	$z_0 - 2057$
	$z_1 - 977$	$z_1 - 2773$
	$z_2 - 2702$	$z_2 - 2305$
відсоток сбіжності	$\sim 44.77\%$	$\sim 61.51\%$
	<i>Схема 2.1</i>	<i>Схема 2.2</i>
		
площі басейнів	$z_0 - 2058$	$z_0 - 2422$
	$z_1 - 3022$	$z_1 - 2192$
	$z_2 - 2407$	$z_2 - 2821$
відсоток сбіжності	$\sim 64.53\%$	$\sim 64.09\%$
	<i>Схема 3.1</i>	<i>Схема 3.2</i>
		
площі басейнів	$z_0 - 1427$	$z_0 - 1944$
	$z_1 - 2870$	$z_1 - 1118$
	$z_2 - 1321$	$z_2 - 3878$
відсоток сбіжності	$\sim 48\%$	$\sim 59.82\%$

Таблиця 3.2: Статистичні характеристики оцінок швидкості збіжності до різних стаціонарних точок для різних схем

<i>Схема</i>	<i>Стаціонарна</i>	<i>Мода</i>	<i>Мин.</i>	<i>Макс.</i>	<i>Среднее.</i>
1.1	z_0	585	2	1580	605.96
	z_1	33	21	88	34.60
	z_2	1602	742	2548	1578.00
1.2	z_0	592	2	1509	603.00
	z_1	1608	807	2818	1573.00
	z_2	33	21	79	33.42
2.1	z_0	920	2	2657	946.02
	z_1	2349	753	3540	2306.00
	z_2	43	27	123	43.99
2.2	z_0	930	2	2699	946.02
	z_1	2345	626	4242	2293.00
	z_2	43	25	105	42.48
3.1	z_0	1845	2	4591	1894.95
	z_1	4690	1795	4997	4573.21
	z_2	90	41	194	93.75
3.2	z_0	1652	2	4439	1874.03
	z_1	4696	2297	5000	92.53
	z_2	91	47	214	4544.60

РОЗДІЛ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНИХ МЕТОДІВ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ НЕЛІНІЙНОЇ ДИНАМІКИ

У данному розділі розроблена програмна реалізація запропонованих методів навчання нейронних мереж на основі нелінійної динаміки та проведено їх апробацію при побудові підсистеми сегментації та класифікації ортопантомограм для стоматологічної інтелектуальної інформаційної системи.

При побудові підсистеми вирішувалась задача перетворення ортопантомограм в одонтограми, які є спеціалізованою формою представлення інформації лікаря про стан кожного зуба пацієнта, з застосуванням згорткової нейронної мережі.

Задача перетворення ортопантомограм в одонтограми знаходиться в фокусі досліджень, а також має великий потенціал використання у лікарській практиці. Це завдання було поставлено, і частково вирішено на замовлення французької компанії ArtifaktAI (Нант, Франція).

Слід зазначити, що реалізація нової модифікації методу градієнтного спуску показала конкурентні переваги як для швидкості навчання, так і для створення одонтограм.

4.1 Перетворення ортопантомограм в одонтограми

У зв'язку зі збільшенням населення нашої планети і часто неможливістю для медичних фахівців високої категорії фізично дістатися до своїх пацієнтів (що в більшості випадків пов'язано з віддаленістю житлового району до найближчого зуболікарського офісу), стає особливо актуальною необхідність частково автоматизувати деякі стандартні процедури. Однією з таких процедур є отримання ортопантомограм та перетворення її на одонтограми. Процедура отримання

ортопантомогра є неінвазивною та швидкою, що здійснюється спеціалізованими апаратами – панорамними рентгенівськими сканерами (рис. 4.1).



Рисунок 4.1: Панорамний рентгенівський сканер

Результатом роботи даних апаратів є цифрові пласкі зображення високої роздільної здатності (рис. 4.2). Один панорамний знімок може надати повну та достовірну інформацію про стан пацієнта для лікаря. Програмне забезпечення сканера дозволяє проводити склеювання зображень по мірі переміщення сканера навколо голови. Таке сканування, природно, призводить до серйозних деформацій. Особливо можна відзначити деформації навколо передніх зубів, де здійснюється проектне перетворення знімка.



Рисунок 4.2: Зразок панорамного знімка зубів (всі дані про власника були видалені до розміщення зображення в Інтернеті). <https://image.jimcdn.com/app/cms/image/transf/none/path/s1fc1121a2c84b65d/image/i7f6f3920c6441383/version/1403060499/digital-x-rays-are-better-for-patients.jpg>

Це призводить до появи світлих вертикальних смуг. Ця обставина обмежує використання існуючих методів машинного навчання та застосування методів обробки зображень для класифікації окремих об'єктів між класами: зуби, пломба, коронка, ясна або кістка. І тому аналіз проводиться, як правило, у ручному режимі. Слід відмітити, що така технологія вимагає додаткових зусиль від медперсоналу та особливо лікаря з аналізу отриманого зображення, та формування одонтограм (рис. 4.3), що містить повну інформацію про стан кожного зуба.

Дослідження, які були проведені компанією Artifakt, показали, що середні витрати на обробку кожного панорамного знімка становлять 10 – 15 хвилин (за умови, що знімок не містить складних та нестандартних ситуацій).

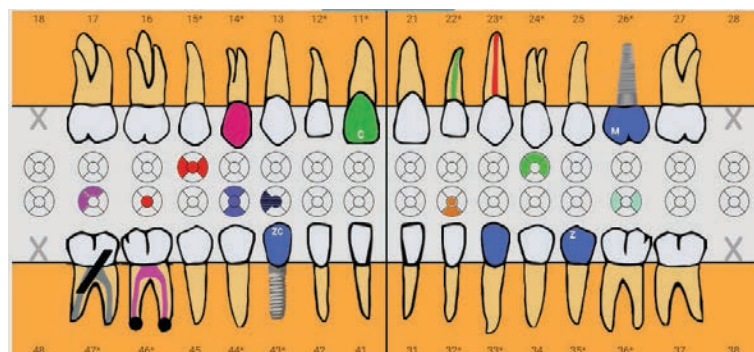


Рисунок 4.3: Приклад одонтограми.

Однак кількість таких знімків, які одержані на протязі року, становить, на-

приклад, у Франції близько 3 млн. Отже, начебто невеликі часові витрати (у десятки хвилин) високооплачуваного фахівця перетворюються на багатомільйонні втрати часу та, відповідно, грошей. Також необхідно врахувати підвищення ймовірності помилкової інтерпретації панорамного знімка та можливих помилок введення даних при опису вигляду одонтограми. Це може призвести до ризику проведення помилкових діагнозів, що в медицині є недопустимим. Таким чином, виникає гостра потреба в автоматизації перетворення ортопантомограм в одонтограми, бо лише автоматизація може зняти відмічені складності.

Крім того, слід відмітити, що крім прямого використання ортопантомограм та результатів їхньої обробки в медичній практиці існує ще одне важливе застосування – ідентифікації особи за станом зубів [74, 75]. Це завдання виникає після різного роду нещасних випадків (автомобільні аварії, аварії літаків та поїздів, пожежі тощо). Відомо, що в найскладніших випадках єдине, на що можуть розраховувати різні служби при пошуку прізвища померлого – це аналіз стану зубів, тому що всі м'які тканини можуть бути знищені. В цьому випадку в багатьох країнах всім зубним лікарям розсилаються електронні повідомлення з описом зубів разом зі знімками (посмертними) для того, щоб кожен зубний лікар спробував впізнати одного зі своїх пацієнтів. Звичайно цей процес дуже повільний, затратний і, практично, повністю ручний. Причина цього часто пов'язана з тим, що, не маючи достатнього часу, лікарі не встигають описати знімок і залишають його у своїй програмі у вигляді зображення. При цьому втрачається зв'язок між двома уявленнями інформації, з яких тільки одонтограма є зручною для індексації, зберігання та швидкого пошуку у різних базах даних.

Таким чином, автоматизація створення бази одонтограм є важливою задачею, яку потрібно вирішити для вирішення важливої задачі для криміналістів –

ідентифікації особи за станом зубів. Це дозволить не тільки скоротити часові та матеріальні витрати на ідентифікацію, а й підвищити її ймовірність.

Проведений аналіз показав, що проблема перетворення ортопантомограм в одонтограми вимагає нового підходу до її вирішення [14, 41, 51, 53, 53, 85, 86, 94]. Таким підходом є застосування методів ІАД і глибоких нейронних мереж для обробки та аналізу медичних даних, у даному випадку рентгенівських знімків. У рамках цього дослідження розроблена підсистема перетворення рентгенівських знімків на одонтограми для стоматологічної інтелектуальної інформаційної системи.



Рисунок 4.4: Архітектура ІАД системи сегментації зубів

Основною складовою, що визначає ефективність підсистеми перетворення ортопантомограм в одонтограм, є навчена глибока нейронна мережа. Для її використання необхідно виконати такі кроки:

- 1) вибрати структуру нейронної мережі;
- 2) сформувати базу ортопантомограм;
- 3) навчити нейронну мережу;
- 4) провести верифікацію.

Розробка та навчання нейронних мереж проводилося на комп'ютері з 64GB оперативної пам'яті та графічним прискорювачем Nvidia Quadro RTX 3000. Даний прискорювач оснащений 6GB відеопам'яті, яку можна використовувати для тренування нейронних мереж. Все програмне забезпечення було створено на основі OpenSource бібліотеки компанії Facebook: PyTorch. Етапи навчання нейронної мережі для тестування різних методів навчання наведено на рисунку 4.5.

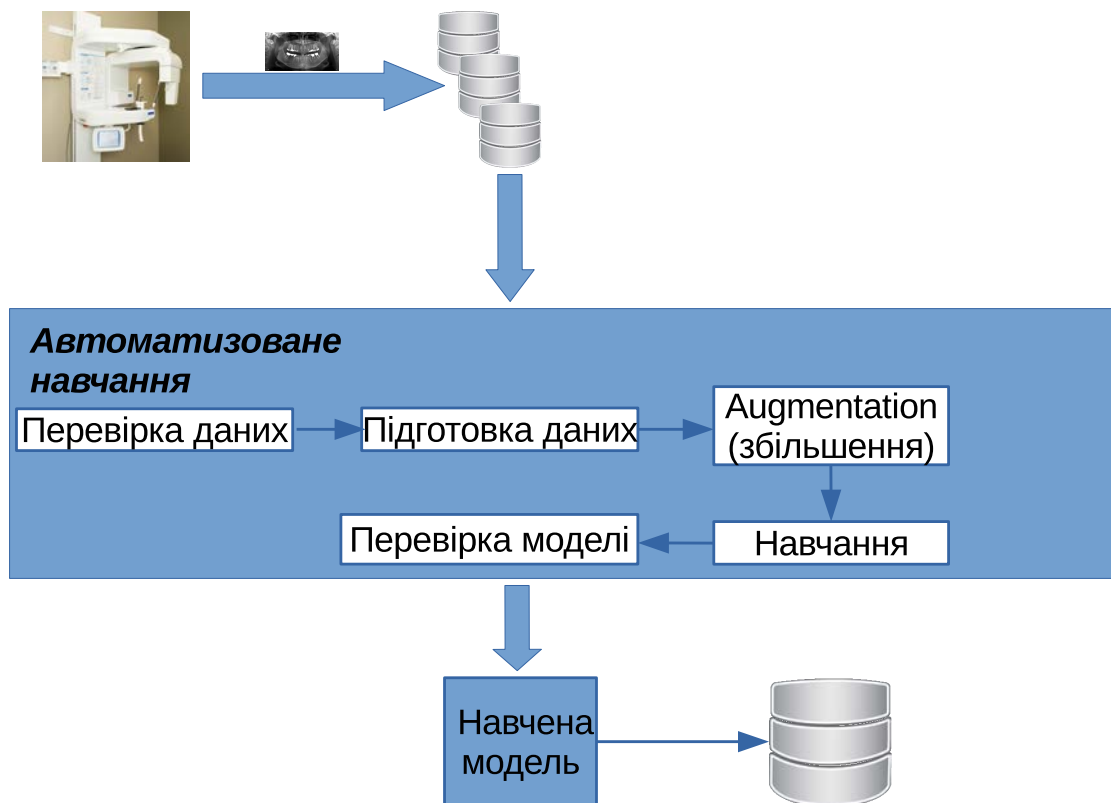


Рисунок 4.5: Етапи навчання нейронної мережі

Процес навчання містить ряд наступних кроків:

- зібрати необхідну кількість панорамних знімків та провести їх розмітку;

- згенерувати множину ортопантомограм, використовуючи методи аугментації даних;
- розбити отриману множину на 3 частини: навчальну, тестову та перевірочну;
- провести навчання нейронних мереж різної архітектури та при різних значеннях їх параметрів;
- обрати найкращу архітектуру нейронної мережі, тобто таку, яка покаже найкращий результат на перевірочному наборі даних.

4.2 Підготовка даних для навчання нейронної мережі

Під час збору інформації виникає низка труднощів, які пов'язані з тим, що в медицині дуже важко та/або неможливо отримати доступ до потрібного обсягу даних. Це пов'язано з такими причинами:

- наявністю малої кількості аналізів чи знімків;
- захистом персональних даних пацієнтів та складністю консолідування даних в одному місці після процесу обробки;
- відсутністю необхідної кількості фахівців (лікарів) для формування навчальної вибірки.

Тому для генерації навчальної вибірки ортопантомограм використано методи аугментації даних.

У роботі початковий обсяг даних було обмежено 54 панорамними зображеннями. Кожне із зображень уявляло собою чорно-біле зображення з 256 градаціями сірого кольору у форматі JPG. Розмірність кожного із зображень була 3000 на 1500 пікселів.

Усі зображення пройшли попереднє очищення від шумів методами обробки зображень та кілька фахівців медиків розмітили зображення, виділивши кожен зуб та відповідну бінарну маску. Кожна маска представляє собою зображення, де інтенсивність кожного пікселя відображає ймовірність приналежності відповідному зубу (значення 1) або фону (значення 0). Приклад маски для одного зі знімків можна побачити на рисунку 4.6.

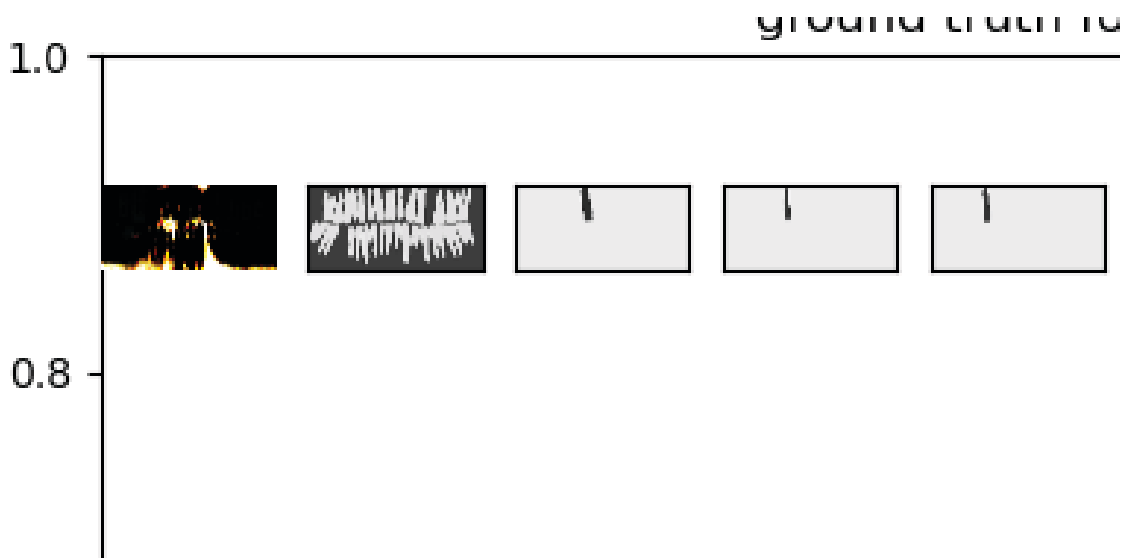


Рисунок 4.6: Панорамний знімок, разом з масками зубів, що підготовлено для навчання нейронної мережі.

На рис. 4.6 у верхньому лівому кутку спочатку розташований панорамний знімок, після нього слідує маска, зібрана з масок всіх зубів - так звана *глобальна маска*.

Вона є інверсною (зуби білі, а фон - чорний). Щоб мережа має навчитися розбивати кожен панорамний рентгенівський знімок не на 32 класи, а на 33. 33 клас - це повна маска всіх зубів. Такий підхід дозволить одразу визначати тканини ротової порожнини без зубів за один прохід через нейронну мережу.

Наступною задачею, що потрібно вирішити при розробці та навчанні нейронної мережі, є генерація множини ортопантомограм на основі даних малого обсягу бази зображень (54 зображення). Отримані множини необхідно розбити на

З окремі підмножини. Перша з них – це підмножина тренувальних зображень, на якому проводиться навчання нейронної мережі, друга – перевірна, яка використовується наприкінці кожної епохи тренування для перевірки результатів навчання на даних, які не входили в цей процес. Остання підмножина використовується лише наприкінці навчання для перевірки точності результату. Зазвичай для верифікації мережі виділяється 20% випадково вибраної підмножини даних.

Оскільки глибокі нейронні мережі мають велику кількість внутрішніх параметрів, їх слід навчати на великому обсязі даних. У нашому випадку нейронна мережа матиме близько 18 мільйонів параметрів, у тому числі близько 50% вимагають навчання. З цього випливає необхідність мати щонайменше півмільйона зображень для навчання нейронної мережі. Отримання такого обсягу панорамних знімків є дуже трудомістким процесом як для наукового співтовариства так і для приватного бізнесу. Тому використовуються методи так званої *аугментації* даних або, іншими словами, *розширення*. В основі цих методів лежать різні геометричні або попиксельні перетворення, які можуть дозволити згенерувати зображення, які схожі на вихідні.

Складність поставленого завдання потребує спеціальних підходів не тільки в процесі аугментування, що дозволить ефективно та повністю використати всі ресурси доступного для проекту обладнання, а й зберігання цього обсягу у вигляді, що дозволяє швидко вибирати дані у процесі навчання.

Для генерації такого великого обсягу даних було розроблено спеціалізоване програмне середовище з можливістю розподілу навантаження між кількома GPU і між різними серверами в кластері.

Перша частина завдання вирішувалася за допомогою поширеної бібліотеки *Albumentations* ([12]). Ця популярна бібліотека увійшла до множини сумісних з

PyTorch бібліотек.

Вибрані для тренування 44 зображення пройшли через наступні деформації: розмиття (функція `Blur`), зміна кольору (функція `ColorJitter`), гаусське розмиття (функція `GaussianBlur`), гаусівське зашумлення (функція `GaussNoise`), спотворення сітки (функція `GriDistor`), горизонтальний зсув (`HorizontalFlip`), зміну насиченості (`HueSaturationValue`), поворот (`Rotate`), еластичне перетворення (`ElasticTransform`), перспективне перетворення (`Perspective`), поворот-масштабування-зсув (`ShiftScaleRotate`).

Всі функції були налаштовані таким чином, щоб випадкові комбінації їхніх параметрів виклику, а також випадковий порядок їхнього застосування створювали панорамні знімки, близькі до реально отриманих з можливими на практиці дефектами і трансформаціями. Насправді, часто трапляються випадки, коли точність позиціонування голови, отже, і якість знімка залежать одночасно як від пацієнта, і від людини, яка проводила дослідження.

Важливо відзначити, що геометричні перетворення проводилися і над знімками і масками зубів, а попиксельні перетворення (наприклад, за допомогою функцій `Blur` і `GaussNoise`) лише над зображеннями.

Всі згенеровані зображення були збережені в одному багатовимірному масиві даних для зниження часу розрахунків та гарантії стабільності надходження даних у нейронну мережу під час її навчання.

Традиційно в реальних умовах нейронні мережі тренуються на випадкових (аугментованих) даних, але в нашому випадку важливо було гарантувати певний порядок для порівняння різних методів оптимізації при навчанні.

Етапи аугментації даних з моменту отримання знімків до створення багатовимірного масиву даних представлено на рисунку 4.7. Отримана множина даних

була розбита на 3 частини: навчальну, тестову та перевірочну.

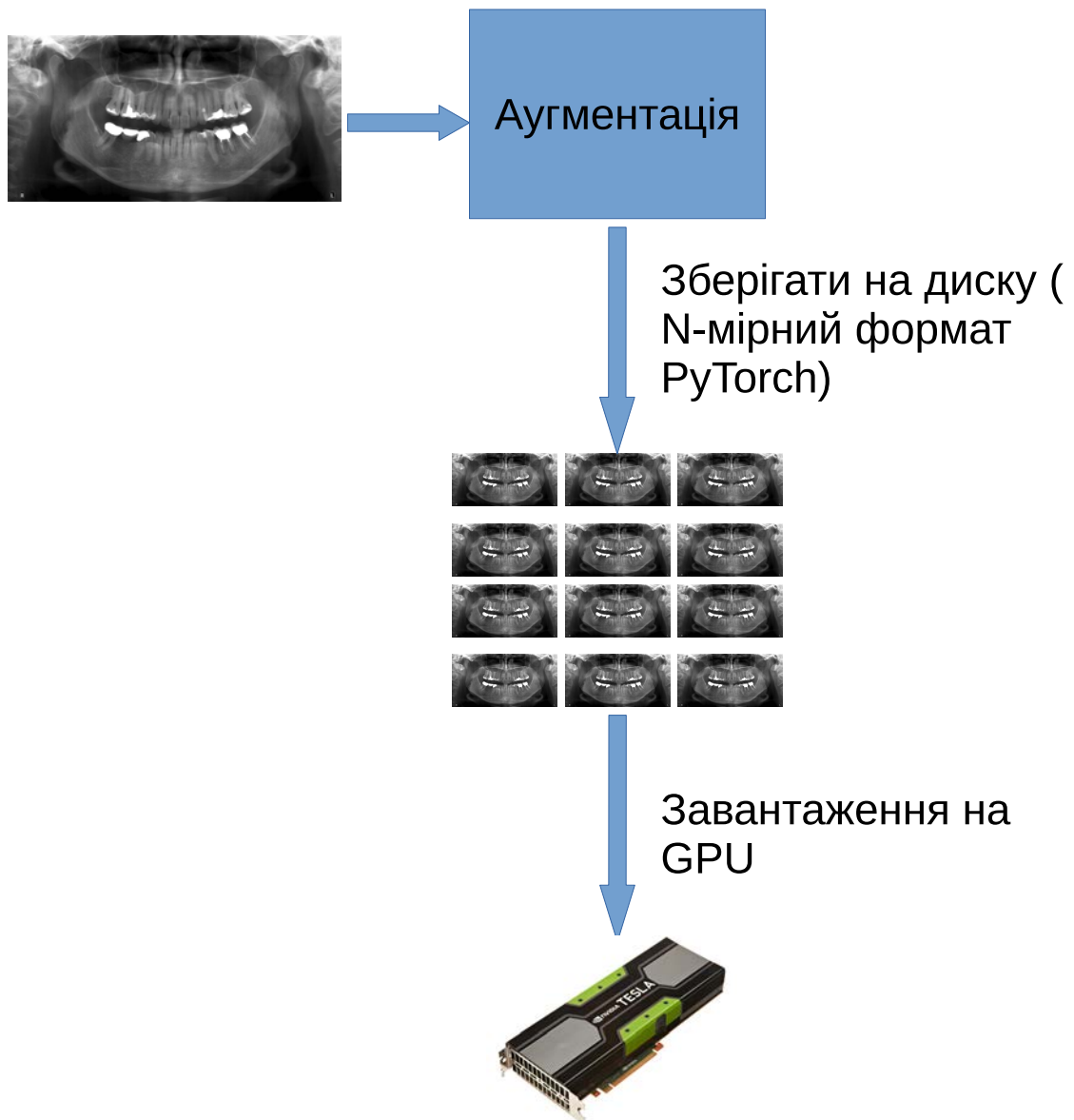


Рисунок 4.7: Етапи процесу аугментації.

4.3 Вибір архітектури нейронної мережі

Невелика кількість даних та специфіка завдання, а саме чорно-білі зображення звужують класи нейронних мереж, готових до використання і спеціально розроблених для роботи з медичними зображеннями. Одна з таких архітектур має назву U-Net ([81]). Розроблена вона Olaf Ronneberger, Philipp Fischer та Thomas Brox спеціально для завдань візуального розпізнавання образів, отриманих на еле-

ктронному мікроскопі. На ній вдалося показати високу швидкість навчання на обмеженому обсязі даних.

Архітектура цієї глибокої нейронної мережі представлена у першому розділі на рисунку 1.6. Ця мережа побудована на основі конволюцій і складається із двох великих частин, з'єднаних посередині. Перша частина архітектури називається *кодувальником*. Ця частина отримує на вході зображення розміром 572 x 572 пікселя та після серії конволюцій повертає набір з 1024 властивостей.

Так само для прискорення навчання та передачі інформації між двома частинами "перекинуто" міст, що з'єднує матриці на рівнях з однаковою глибиною з боку *кодувальника* та *декодувальника*.

Для прискорення процесу навчання та збільшення точності було випробувано кілька підходів до вибору архітектури нейронної мережі. Перший, з використанням чистої архітектури U-Net і другий, із заміною кодувальника, який виділяє важливі геометричні та властивості інтенсивності зображення. В ролі кодувальника був використаний кодувальник з іншої архітектури ResNet18 [55]. Цей підхід був обраний через те, що завдання кодувальника полягає у виділенні із зображення ключових характеристик, таких як межі об'єктів, їхні форми та розташування на зображенні. А для ResNet існує попередньо навчена глибока нейронна мережа зображень ImageNet [52], яка містить близько тисячі об'єктів та десятки мільйонів зображень з ними. Навчання ResNet на такому великому обсязі даних займає за тривалістю тиждень на звичайному обладнанні, а для того, щоб отримати результат за кілька годин або за годину, потрібний великий кластер серверів і програмне забезпечення, спеціально створене для цього випадку. Але оскільки всі популярні бібліотеки глибокого навчання містять попередньо навчену мережу ResNet, ми можемо використовувати навчені кодувальники.

Таким чином, вдалося використати найкращі якості двох архітектур, пристосувати досвід, отриманий під час аналізу зображень, у медичній галузі. Але головну перевагу в цій роботі приніс не оригінальний архітектурний підхід, а розробка нового методу оптимізації при навчанні.

Після підготовки вихідних даних основна частина системи будувалася на базі нової бібліотеки Pytorch Lightning [36], розробленої на базі PyTorch для полегшення розробки та досліджень у глибокому навчанні. Перевага даної бібліотеки полягає в її можливості розпаралелювати тренувальний процес не лише між декількома ядрами одного процесора та/або декількома графічними прискорювачами на одному сервері, але й використовувати різні архітектури розподілених обчислень (включаючи "хмарні" рішення Amazon, Azure та Google).

Оскільки бібліотека Pytorch Lightning сама є розширенням PyTorch, необхідні для даного дослідження розширення методів оптимізації десь як існуючий алгоритм бажано проводити лише на рівні абстракції PyTorch. Ця бібліотека має спеціальний інтерфейс `torch.optim.Optimizer`, який повинен реалізувати всі методи, що підтримуються.

4.4 Навчання нейронної мережі

Процес навчання проводився на комп'ютері з вбудованим GPU та займав у середньому близько однієї години для кожного оптимізатора. Щоб отримати початкове уявлення про особливості навчання для даної архітектури нейронної мережі та згенерованих навчальних даних, було застосовано рекомендований алгоритм Adam з коефіцієнтом навчання рівним 0.0001. Декілька запусків для 100 епох показали, що мережа припиняє навчатися досить швидко, а значить, не має сенсу витрачати стільки часу та ресурсів, тому було запропоновано обмежитися

для порівняння на 50 епохами.

Більшість бібліотек розробки алгоритмів машинного навчання використовують генератори псеводо-випадкових чисел на різних етапах. Наприклад, кожному епоху із загального набору панорамних знімків вибираються 15 (15 – в даному випадку встановлений розмір batch, що пов'язано з розмірами зображень та можливостями графічного прискорювача, особливо розміру оперативної пам'яті цього прискорювача, рівне 6GB) для панорамних зображень розміру 224 на 224. Цей розмір був обмежувачем і важливим параметром, оскільки частина мережі (кодер), вже було навчена на зображеннях такого розміру.

Генерація 2000 зображень з невеликого навчального набору призводила до стабілізації процесу навчання після 5 епох. Ще 5 додавалися, на випадок, якщо поруч з локальним мінімумом знаходиться плоска ділянка або один з алгоритмів застряг у проваллі. Залишалася можливість перебувати на площині або в проваллі тривалий час, але обмежені ресурси та час не дозволяли очікувати надто тривале покращення точності. Навчання примусово зупинялося у тому випадку, коли не відбувалося помітних покращень протягом 5 епох.

Результати порівняння роботи різних алгоритмів оптимізації продемонстровані на рисунку (4.8).

Аналіз результатів показав, що найкращі результати навчання були показані алгоритмами Adagrad та Adam (0.0502 та 0.0474 відповідно). Причому Adam як найпоширеніший на сьогоднішній день з існуючих алгоритмів "за замовчуванням" і рекомендований для початкового наближення до локального екстремуму перевершив Adagrad протягом всього навчання і тільки біля 10 епох обидва алгоритми наблизилися один до одного за результатами.

З іншого боку помітна перевага цих двох алгоритмів у порівнянні з SGD (з

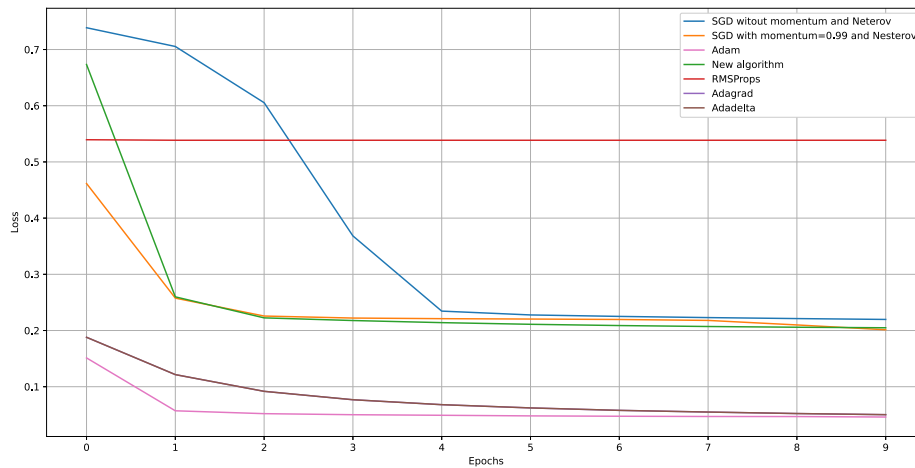


Рисунок 4.8: Результати тренування нейронної мережі з поширеними різновидами градієнтного спуску в порівнянні з розробленим алгоритмом на основі нелінійної динаміки (1.5).

моментом і Несторовим і без них) та запропонованною модифікацією градієнтного спуску.

Так само можна підкреслити явну перевагу у швидкодії нового алгоритму щодо SGD без моменту і Нестерова і його невелике відставання від SGD з моментом і Несторовим на самому початку навчання з наступною невеликою перевагою.

З цього випливає, що нова модифікація за формулою (1.5) з визначеними коефіцієнтами вже між першою та другою епохами досягла кращих значень функції втрати, ніж ті, яких було досягнуто SGD лише на 4 епосі. Це вказує на подвійну перевагу за часом навчання (швидкодією).

У процесі навчання збиралася додаткова інформація, яка дозволяє оцінити реальне використання ресурсів GPU для різних алгоритмів оптимізації без оцінки складності вихідного коду самих алгоритмів. Результати визначення похибки мінімуму функції втрати (точності, досягнутої оптимізатором при навчанні) та значення функції втрати (точності) на тестових даних представлено у таблиці 4.1, у якій жирним шрифтом позначені кращі результати щодо кожного з вимірів.

Аналіз даних, що наведені у таблиці показує, що новий алгоритм (1.5) не сильно поступився показниками використання оперативної пам'яті GPU і витратив зайві 21 мегабайта.

Мінімальне значення функції втрати було досягнуто алгоритмом

Таблиця 4.1: Таблиця порівняння результатів різних алгоритмів оптимізації та споживання ресурсів GPU

Алгоритм	GPU пам'ять (MiB)	Мінімуму	Тестові результати
RMSProps	3503	1.3766	0.5385
Adagrad	3571	0.0502	0.5269
Adadelta	3459	0.5520	0.5560
Adam	3459	0.0474	1.189
SGD	3450	0.2197	0.5365
Формула (3)	3471	0.2673	0.2049

Adam, але найкращий, її результат на валідаційному (тестовому) наборі даних було отримано розробленим алгоритмом градієнтного спуску.

Останній цікавий параметр обчислювальної складності – кількість ітерацій за одиницю часу. За цим параметром має сенс оцінити лише три алгоритми: SGD, Adam та новий алгоритм. Бібліотека Pytorch Lightning наприкінці кожного запуску показує швидкість навчання. Для трьох аналізованих алгоритмів ці значення становлять 38.55 it/s, 39.0 it/s та 39.42 it/s, відповідно. З цього випливає, що навіть враховуючи додаткове обчислення градієнта функції втрати, новий алгоритм дозволяє процесу навчання мати таку ж швидкість, як і інші алгоритми.

Отримані результати дозволили прискорити процеси навчання ряду архітектур нейронних мереж, котрі розроблені як частина програмного забезпечення компаній Artefakt AI. Таке прискорення дозволило здійснити широкий пошук оптимальних гіперпараметрів для кожної з мережевих архітектур, використання обмежених ресурсів обчислювального кластера.

Отримано 20% прискорення процесу навчання (див. Додаток А), і в резуль-

таті нейронна мережа зможе працювати як ядро особистого додатка, так і в якості основної частини апаратного розширення для рентгенівських сканерів.

4.5 Висновки до четвертого розділу

Застосування нової модифікації градієнтного спуску у процесі навчання нейронних мереж різних класів та розмірів продемонструвало його переваги щодо алгоритму SGD.

Застосування методів ІАД для вирішення задачі сегментації панорамних рентгенівських знімків зубів на підприємстві дозволило вдвічі (2 епохи порівняно з 4 епохами для SGD) скоротити час навчання. Новизна рішення складалася не тільки з нових методів оптимізації, але й методів попередньої обробки, зберігання та аугментації прикладів, яких не вистачало для навчання. Усе це вимагало прийняття багатьох технологічних рішень та використання всіх доступних апаратних засобів. Серед побічних результатів які були одержані є удосконалення процеси генерації навчальних даних та оптимізація використання пам'яті GPU. Це дозволило прискорити процес навчання у 10 разів (з 50 секунд на один пакет даних до 5 секунд). Висвободжена додаткова пам'ять дозволила збільшити розмір кожного пакета даних з 7 зображень на пакет до 15. Це призвело до меншого коливання градієнтів та більш швидкої збіжності до локального мінімуму з рахунком згладження траєкторії пошуку.

Практичним результатом є НМ та можливість масштабувати рішення на інші застосування.

Проте, найважливішим показником, за яким нова модифікація градієнтного спуску перевершила всі модифікації SGD і навіть Adam, є величина помилки передбачень мережі на верифікаційному (тестовому) наборі даних. Це найважли-

віший параметр, за яким оцінюється точність налаштування мережі. За цим параметром новий алгоритм у 2 рази перевершив SGD та у 5.8 рази – Adam. Для медичних завдань, де вимога точності рішень і найкраще наближення до локального мінімуму цільової функції, є пріоритетним завданням, це є важливішим, ніж швидкість навчання мережі.

Ці результати доводять не лише практичну застосовність нових рішень, які запропоновані у даній роботі для задачі автоматичного створення одонтограми, а й показують, великі перспективи дослідження для вирішення інших завдань ІАД, в основі яких знаходяться нейронні мережі та, як наслідок, великі обсяги даних.

ЗАГАЛЬНІ ВИСНОВКИ

Сукупність отриманих в дисертації результатів вирішує актуальну науково-технічну задачу скорочення часу навчання нейронних мереж під час вирішення завдань сегментації та класифікації.

Після проведених досліджень отримано наступні наукові результати, які є істотним внеском у подальше удосконалення методів градієнтного спуску та пошуку його параметрів.

1. Виконано аналіз існуючих нейронних мереж для вирішення завдань сегментації та класифікації зображень, що дозволило обрати та обґрунтувати вибір напрямку досліджень щодо побудови нових удосконалених градієнтних методів які можуть скоротити час навчання та точність результатів; виконано аналіз сучасного стану методів градієнтного спуску, які застосовуються в індустрії при вирішенні практичних завдань сегментації та класифікації.

2. Удосконалений метод градієнтного спуску на основі положень нелінійної динаміки. На його базі розроблені модифікації методу з урахуванням зворотного зв'язку з запізненням та прогнозуючого зворотного зв'язку в ДДС. Ці методи налаштовуються за допомогою безлічі параметрів, які можуть змінюватися в залежності від оптимізаційної задачі.

3. Удосконалені методи навчання для різних архітектур НМ градієнтного спуску та оптимізаційних задач.

4. Розроблений метод пошуку параметрів для запропонованих модифікацій градієнтного спуску на основі розв'язання оптимізаційних задач на класі комплексних поліномів. Отримані теоретичні результати були випробувані при розв'язанні різних класів оптимізаційних завдань, у яких використовується градієнтний метод.

5. Реалізовано розширення бібліотеки глибокого навчання PyTorch новими модифікаціями методу градієнтного спуску та показано їх перевагу при навчанні багаторівневого персептрона відносно стандартного методу градієнтного спуску.

6. Нові методи були впроваджені в навчання глибокої нейронної мережі, створеної для сегментації та класифікації зубів у компанії Artifakt AI (Франція), де одержали 20% відсоткове прискорення у навчанні нейронної мережі

Результати досліджень впроваджені в навчальному процесі ОНПУ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Andrievskii V.V., Ruscheweyh S. Maximal polynomial subordination to univalent functions in the unit disk, *Constr. Approx* 10, 1994, P. 131-144. <https://doi.org/10.1007/BF01205171>
- [2] Beliaev D., Smirnov S. Harmonic measure on fractal sets, in *European Congress of Mathematics*, Eur. Math. Soc., Zürich, 2005. P. 41-59
- [3] Beliaev D., Smirnov S. Random conformal snowflakes. *Annals of Math.* 172, 2010. No 1. P. 597-615.
- [4] Bieberbach L. Über die Koeffizienten derjenigen Potenzreihen, welche eine schlichte. Abbildung des Einheitskreises vermitteln. S.-B. Preuss. Akad. Wiss. 138, 1916. P. 940-955.
- [5] Boas H.P., Khavinson D. Bohr's power series theorem in several variables. *Proc. Amer. Math. Soc.* 125, 1997. P. 2975-2979.
- [6] Bohr H. A theorem concerning power series. *Proc. London Math. Soc.* (2) 13, 1914. P. 1-5.
- [7] Brandt M. Variationsmethoden für in der Einheitskreisscheibe schlichte Polynome. Dissertation. Humboldt-Univ. Berlin. 1987
- [8] Brandt M. Representation formulas for the class of typically real polynomials, *Math. Nachr.* 144, 1989. P. 29-37.
- [9] Brandt M., *Variationsmethoden für in der Einheitskreisscheibe schlichte Polynome*, Dissertation, Humboldt-Univ. Berlin, 1987.
- [10] Brannan D.A. On univalent polynomials, *Glasgow Math. J.* 11, 1970. P. 102-107.

- [11] Brannan D.A. Coefficient regions for univalent polynomials of small degree. *Mathematika* 14, 1967. 165-169
- [12] Buslaev, A., Iglovikov A., Vladimir I., Khvedchenya, Eugene and Parinov, Druzhinin A., Kalinin M., Alexandr A., *Albumentations: Fast and Flexible Image Augmentations*, *Information*, 11, 2020, 2, 125, URL: <https://www.mdpi.com/2078-2489/11/2/125>, DOI: 10.3390/info11020125
- [13] Carleson L., Jones P. W. On coefficient problems for univalent functions and conformal dimension, *Duke Math. J.* 66, 1992. P. 169-206.
- [14] Chen H., Zhang K., Lyu P., Li H., Zhang L., Wu J., Lee C. A deep learning approach to automatic teeth detection and numbering based on object detection in dental periapical films. *Scientific Reports*, 2019. V. 9, № 3840, DOI:<https://doi.org/10.1038/s41598-019-40414-y>
- [15] Chu C. Asymptotic Bohr radius for the polynomials in one complex variable, in: *Invariant subspaces of the shift operator*, *Contemp. Math.* 638, Amer. Math. Soc., Providence, RI, 2015. P. 39-43.
- [16] Cordova A., Ruscheweyh St. On maximal ranges of polynomial spaces in the unit disk, *Constr. Approx.* 5, 1989. P. 309-328.
- [17] Cowling, V.F., Royster, W.C. Domains of variability for univalent polynomials. *Proc. Am. Math. Soc.* 19, 1968. P. 767-772
- [18] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego, 2015. DOI: <https://doi.org/10.48550/arXiv.1412.6980>

- [19] Dillies J., Dmitrishin D., Stokolos A. On Suffridge Polynomials, 2020. URL: <https://arxiv.org/abs/2007.09255>
- [20] Dimitrov D. Extremal univalent polynomials subordinating the Koebe function, East journal on approximations, 2005. Vol. 11, № 1. P. 47-56.
- [21] Dimitrov D., Extremal positive trigonometric polynomials, in: Approximation theory: A volume dedicated to Blagovest Sendov (B. Bojanov, ed.), 2002. P. 1-24.
- [22] Dmitrishin D., Dyakonov K., Stokolos A. Univalent polynomials and Koebe's one quarter theorem. Analysis and Mathematical Physics, 2019. V. 9. P. 991-1004
- [23] Dmitrishin D., Dyakonov K., Stokolos A., Univalent polynomials and Koebe's one-quarter theorem, Anal. Math. Phys, 2019. V.9. P. 991-1004. DOI:<https://doi.org/10.1007/s13324-019-00305-x>
- [24] Dmitrishin D., Hagelstein P., Khamitova A., Korenovskyi A., Stokolos A. *Fejér Polynomials and Control of Nonlinear Discrete Systems*, Constructive Approximation 51, 2020. P. 383-412.
- [25] Dmitrishin D., Hagelstein P., Khamitova A., Stokolos A. Further stability results for a generalization of delayed feedback control. Nonlinear Dynamics, 2012. P. 1-8.
- [26] Dmitrishin D., Hagelstein P., Khamitova A., Stokolos A. Limitations of Robust Stability of a Linear Delayed Feedback Control, SIAM J. Control Optim. 56, 2018, P. 148-157.
- [27] Dmitrishin D., Iacob, I., Skrinnik, I., Stokolos, A. Finding, Stabilizing, and Verifying Cycles of Nonlinear Dynamical Systems, 2017. URL:<https://arxiv.org/abs/1712.06035>

- [28] Dmitrishin D., Khamitova A. Methods of harmonic analysis in nonlinear dynamics, *Comptes Rendus Mathematique*, 2013. V. 351. I. 9-10. P. 367 -370
- [29] Dmitrishin D., Khamitova A., Stokolos A., Tohaneanu M. Finding Cycles in Nonlinear Autonomous Discrete Dynamical Systems. *Harmonic Analysis, Partial Differential Equations, Banach Spaces, and Operator Theory (Volume 2)*, Springer AWM series, 2017. V. 5. P. 199-237.
- [30] Dmitrishin D., Khamitova A., Stokolos A.M. Fejér Polynomials and Chaos. , and Harmonic Analysis. *Springer Proceedings in Mathematics & Statistics*, 2014. V. 108. P. 49-75.
- [31] Dmitrishin D., Smorodin A., Stokolos A. On a family of extremal polynomials. *Comptes Rendus Mathematique*, 2019. V. 357. I. 7. P. 591-596.
- [32] Dmitrishin D., Smorodin A., Stokolos A. Estimating the Koebe radius for polynomials. *arXiv*, 2018. DOI:10.48550/ARXIV.1805.06927. URL:<https://doi.org/10.48550/arXiv.1805.06927>
- [33] Dmitrishin D., Stokolos A., Tohaneanu M. Searching for cycles in non-linear autonomous discrete dynamical systems. *New York J. Math.* 25, 2019. P. 603-626
- [34] Duchi J., Hazan E., Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *Journal of Machine Learning Research*, 2011. 12(61), P. 2121-2159. URL:<http://jmlr.org/papers/v12/duchi11a.html>
(дата звернення: 17.04.2022)
- [35] Duren P. L. Univalent Functions, *Grundlehren der mathematischen Wissenschaften*. Springer Science+Business Media. Berlin, 1983. 382 с.

- [36] The lightweight PyTorch wrapper for high-performance AI research. URL:<https://github.com/PyTorchLightning/pytorch-lightning>. (дата звернення: 17.04.2022)
- [37] Fejér L. Ueber trigonometrische Polynome. J. Reine Angew. Math. 146, 1915. P. 53-82.
- [38] Fekete M., Szegő G. Eine Bemerkung über ungerade schlichte Funktionen. J. London Math. Soc, 1933. V. 8. P. 85-89
- [39] Fournier R. Asymptotics of the Bohr radius for polynomials of fixed degree. J. Math. Anal. Appl. 338, 2008. P. 1100-1107.
- [40] Genthner C., Ruscheweyh St., Salinas L. A Criterion for Quasi-Simple Plane Curves. Computational Methods and Function Theory, 2003. V. 2. I. 1. P. 281-291.
- [41] Giger L.M. Machine Learning in Medical Imaging. Journal of the American College of Radiology, 2018, V. 15(3, Part B). P. 512-520. URL:<https://doi.org/10.1016/j.jacr.2017.12.028> (дата звернення: 17.04.2022).
- [42] SGirshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition, 2014, P. 580-587. DOI: 10.1109/CVPR.2014.81.
- [43] Gradshtejn I.S., Ryzhik I.M. Tablicy integralov, summ, rjadov i proizvedenij. 4-e izd. M. Fizmatgiz, 1963. 1100 c.
- [44] Greiner R. On a theorem of Andrievskii and Ruscheweyh, Proceedings of the

Ashkelon Workshop on Complex Function Theory. Israel Math. Conf. Proc, 1996, P. 83-90

- [45] Greiner R., Ruscheweyh St. On the approximation of univalent functions by subordinate polynomials in the unit disc. Approximation and Computation. Internat. Ser. Numer. Math. 119, Birkhuser, 1994. P. 261-271. DOI:10.1007/978-1-4684-7415-2_16 URL:https://doi.org/10.1007/978-1-4684-7415-2_16 (дата звернення: 17.04.2022)
- [46] Guadarrama Z. Bohr's radius for polynomials in one complex variable. Comput. Methods Funct. Theory 5, 2005. P.143-151.
- [47] Hayman W. Multivalent functions. Cambridge University Press, N. Y., 1994. 276 p. ISBN: 9780521460262
- [48] Hayman W. K. The asymptotic behaviour of p-valent functions, Proceedings of the London Mathematical Society, Third Series, 5 (3), 1955. P. 257-284, DOI:10.1112/plms/s3-5.3.257
- [49] Hinton G.E., Shallice T. Lesioning an attractor network: investigations of acquired dyslexia. Psychological review, 1991, 98(1), p. 74-95. DOI:10.1037/0033-295X.98.1.74
- [50] Hinton, G. E., Osindero, S., Teh, Y. A fast learning algorithm for deep belief nets. Neural Computation 18(7), 2006. P. 1527-1554. DOI:10.1162/neco.2006.18.7.1527. PMID: 16764513.
- [51] Hwang JJ, Jung YH, Cho BH, Heo MS. An overview of deep learning in the field of dentistry. Imaging Sci Dent 49(1), 2019. P. 1-7. DOI:<https://doi.org/10.5624/isd.2019.49.1.1>

- [52] ImageNet. URL:<http://www.image-net.org> (дата звернення: 17.04.2022).
- [53] Jader G. and others. Deep Instance Segmentation of Teeth in Panoramic X-Ray Images, 2018, P. 400-407. DOI:10.1109/SIBGRAP.2018.00058
- [54] Jouppi N., Young C., Patil N. In-datacenter performance analysis of a tensor processing unit. 44th International Symposium on Computer Architecture (ISCA), 2017: P. 1–17.
- [55] Kaiming H., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. P. 770-778.
- [56] Kayumov I. R., Khammatova D. M. On a conjecture about univalent polynomials, 2020. arXiv:2002.12304. URL:<https://doi.org/10.48550/arXiv.2002.12304> (дата звернення: 17.04.2022)
- [57] Khalil H. K. Nonlinear Systems, 3rd edn. Prentice-Hall, Upper Saddle River, 2002, 750 p.
- [58] Koebe P. Über die Uniformisierung beliebiger analytischer Kurven. Nachr. Akad. Wiss. Göttingen Math. Phys. Kl, 1907. P. 191-210.
- [59] Krizhevsky A., Sutskever I., Hinton E. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2012. V. 60, P. 84-90.
- [60] Fejér L. Über trigonometrische Polynome, J. Reine Angew. Math. 146, 1916. P. 53-82.

- [61] Lemaréchal C. Cauchy and the Gradient Method. Doc Math Extra, 2012. P. 251–254. URL:https://www.math.uni-bielefeld.de/documenta/vol-ismp/40_lemarechal-claude.pdf (дата звернення: 17.04.2022)
- [62] Lewin W. I. Ein Beitrag zum Koeffizientenproblem der schlichten Funktionen. Math. Z., 1934. V. 38, P. 306-311.
- [63] Littlewood J.E, Paley R. E. A. C. A Proof That An Odd schlicht Function Has Bounded Coefficients. Journal of the London Mathematical Society, 1932. V. s1-7. I. 3. P. 167-169. URL:<https://doi.org/10.1112/jlms/s17.3.167> (дата звернення: 17.04.2022)
- [64] Littlewood J.E. On the coefficients of schlicht functions. Quart. J. Math., 1938. V. 9, No 33, P. 14-20.
- [65] Shelhamer E, Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans Pattern Anal Mach Intell, 2017. 39(4). P. 640-651. DOI: 10.1109/TPAMI.2016.2572683. PMID: 27244717.
- [66] Longsine D.E., McCormick S.F. Simultaneous Rayleigh-quotient minimization methods for $Ax = \lambda Bx$. Linear Algebra Appl. 34, 1980. P. 195-234. URL:<https://www.sciencedirect.com/science/article/pii/0024379580901664> (дата звернення: 17.04.2022)
- [67] Makarov N.G. Fine structure of harmonic measure. St. Petersburg Math. J., 1999. V. 10(2). P. 217-268.
- [68] Marblestone A., Wayne G., Kording K. Toward an Integration of Deep Learning and Neuroscience. Frontiers in Computational Neuroscience. 10: 94,

2016. arXiv:1606.03813. DOI:10.3389/fncom.2016.00094. PMC 5021692. PMID 27683554. S2CID 1994856.
- [69] Martin R.S. and Wilkinson J.H., Reduction of the symmetric eigenproblem $Ax = \lambda Bx$ and related problems to standard form. Numer. Math. 11, 1968. P. 99-110.
- [70] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method, 2012. DOI:10.48550/ARXIV.1212.5701. URL:<https://arxiv.org/abs/1212.5701> (дата звернення: 17.04.2022)
- [71] Michel C., Untersuchungen zum Koeffizientenproblem bei schlichten Polynomen. Dissertation. Humboldt-Univ. Berlin, 1971.
- [72] Morgul O. Further stability results for a generalization of delayed feedback control. Nonlinear Dynamics, 2012. P. 1–8.
- [73] Nesterov Y. A method of solving a convex program-ming problem with convergence rate $O(1/\sqrt{k})$. Soviet Mathematics Doklady, 27, 1983. P. 372–376.
- [74] Nomir O., Abdel-Mottaleb M., Pattern Recognition, 2005. V. 38(8), P. 1295-1305. DOI:<https://doi.org/10.1016/j.patcog.2004.12.010> (дата звернення: 17.04.2022).
- [75] Parvez Mahdi F., Motoki K., Kobashi S. Optimization technique combined with deep learning method for teeth recognition in dental panoramic radiographs. Scientific Reports volume 10, 2020. № 19261. DOI:<https://doi.org/10.1038/s41598-020-75887-9> (дата звернення: 17.04.2022).
- [76] Polyak B.T., Juditsky A.B. Acceleration of Stochastic Approximation by Averaging. Society for Industrial and Applied Mathematics, SIAM J. Control Optim., 30(4), 1992. P. 838–855.

- [77] Polyak, B.T. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics 4(5), 1964. P. 1–17.
- [78] Riedmiller M., Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. IEEE International Conference on Neural Networks , 1993. V. 1. P.586-591. URL:<https://ieeexplore.ieee.org/document/298623> (дата звернення: 17.04.2022)
- [79] Rogosinski W. W. and Szegő G. Extremum problems for non-negative sine polynomials. Acta Sci. Math. (SDCAFzeged) 12, 1950. P. 112–124.
- [80] Rogosinski W. W. Uber positive harmonische sinusentwicklungen. Jber. Deutsch. Math. Ver., 40, 1931. P. 33–35.
- [81] Ronneberger. O., Fischer. F., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2018. DOI:10.48550/ARXIV.1505.04597 URL:<https://arxiv.org/abs/1505.04597> (дата звернення: 17.04.2022).
- [82] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev, 1958. V. 65(6). P. 386-408. DOI: 10.1037/h0042519. PMID: 13602029.
- [83] Rumelhart D. E., Hinton G. E., Ronald W. J. Learning representations by back-propagating errors. Nature 323, 1986. P. 533–536. DOI:10.1038/323533a0. URL:<https://www.nature.com/articles/323533a0> (дата звернення: 17.04.2022).
- [84] Schwartz R., Dodge J., Smith N., Etzioni O. Green AI. Communications of the ACM, 2020, V. 63. №. 12, P. 54-63. DOI:10.1145/3381831. URL:<https://arxiv.org/pdf/1907.10597.pdf> (дата звернення: 17.04.2022).

- [85] Silva G., Oliveira L., Pithon M., Automatic segmenting teeth in X-ray images: Trends, a novel data set, benchmarking and future perspectives., *Expert Systems with Applications*, 2018, 107, 15-31, DOI:<https://doi.org/10.1016/j.eswa.2018.04.001>
- [86] Silva B., Pinheiro L., Oliveira L., Pithon M. A study on tooth segmentation and numbering using end-to-end deep neural networks. 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020. P. 164-171, DOI: 10.1109/SIBGRAPI51738.2020.00030.
- [87] Silver D., Schrittwieser J., Simonyan K. Mastering the game of Go without human knowledge. *Nature* 550, 2017. P. 354–359. DOI:<https://doi.org/10.1038/nature24270> (дата звернення: 17.04.2022).
- [88] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. DOI:10.48550/ARXIV.1409.1556. URL:<https://arxiv.org/abs/1409.1556> (дата звернення: 17.04.2022).
- [89] Simionescu P.A., Smith M.R. Single–Valued Functions Graphical Representations in Linkage Mechanisms Design. *Mech. Mach. Theory*, 2000, V. 35, P. 1709–1726. DOI:10.1016/S0094-114X(00)00018-5. URL:https://www.researchgate.net/publication/223690593_Single-valued_function_representations_in_linkage_mechanisms_design (дата звернення: 17.04.2022).
- [90] Strang G. *Linear algebra and its applications*, 4th ed. Cengage Learning, Boston, 2006.
- [91] Suffridge T.J. On univalent polynomials. *J. London Math. Soc.* 44, 1969. P. 496-504.

- [92] Sutskever I., Martens J., Dahl G., Hinton G. On the importance of initialization and momentum in deep learning. Proceedings of the 30th International Conference on Machine Learning, 2013. V. 28. № 3. P. 1139-1147. URL:<http://proceedings.mlr.press/v28/sutskever13.pdf> (дата звернення: 17.04.2022).
- [93] Valueva M.V., Nagornov N.N., Lyakhov P.A., Valuev G.V., Chervyakov N.I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Mathematics and Computers in Simulation. Elsevier, 2020. V. 177. P. 232–243. DOI:10.1016/j.matcom.2020.04.031. ISSN 0378-4754. S2CID 218955622.
- [94] Wang C. and others. A benchmark for comparison of dental radiography analysis algorithms., Medical Image Analysis, 2016. V. 31, P. 63-76. DOI:<https://doi.org/10.1016/j.media.2016.02.004> (дата звернення: 17.04.2022). URL:<https://www.sciencedirect.com/science/article/pii/S1361841516000190> (дата звернення: 17.04.2022).
- [95] https://en.wikipedia.org/wiki/Digamma_function (дата звернення: 17.04.2022).
- [96] de Branges L. A proof of the Bieberbach conjecture, Acta Mathematica, 1985. V. 154(1). P. 137-152.
- [97] Schooten F. Van. Francisci Vietae Opera Mathematica. Leyde, Elzévir, 1646, p. 554 Hildesheim-New-York: Georg Olms Verlag
- [98] Böttcher A., Grudsky S. M. Toeplitz Matrices, Asymptotic Linear Algebra, and Functional Analysis. Birkhäuser, 2012. ISBN 978-3-0348-8395-5
- [99] Rosenbrock function. URL:<https://upload.wikimedia.org/wikipedia/c>

ommons/thumb/3/32/Rosenbrock_function.svg/2560px-Rosenbrock_function.svg.png (дата звернення: 17.04.2022).

ДОДАТОК А. Відомості про апробацію результатів дисертації



ARTEFAKT AI
Siren: 823 773 387
R.C.S Nantes, France

23.06.2021

«Затверджую»
директор ARTEFAKT AI
Natalia de La Fournière

АКТ ВПРОВАДЖЕННЯ наукових результатів дисертаційної роботи Смородіна А. В. на тему «Методи навчання нейронних мереж на основі нелінійної динаміки»

Цей акт є підтвердженням використання результатів досліджень Смородина А. В., які були отримані в рамках дисертаційної роботи на здобуття наукового ступеня доктора філософії на тему «Методи навчання нейронних мереж на основі нелінійної динаміки».

В період з 02.01.2020 р. до 26.02.2021 р. в компанії Artefakt AI проведено комплекс досліджень з використанням алгоритму, який було розроблено аспірантом Державного університету «Одеська політехніка» Смородіним А.В. в процесі роботи над дисертацією на здобуття наукового ступеня доктора філософії. Даний алгоритм використовувався для навчання нейронних мереж різної архітектури, які компанія Artefakt AI використовує для побудови зубних одонтограм з 2-D мірних рентгенівських панорамних знімків зубів. Нейронні мережі, навчені за допомогою нового методу оптимізації, який був розроблений аспірантом Смородіним А.В., будуть застосовані як вбудовані модулі для зубних програм і апаратних рентгенівських пристроїв.

В результаті використання нового алгоритму вдалося отримати 20% прискорення процесу навчання щодо алгоритму stochastic gradient descent, що дозволило скоротити час тренування нейронних мереж і зменшити ціну використаних кластерних рішень. За цей же час навчання отримані нейронні мережі з більш високими передбаченими здібностями.

Від ARTEFAKT AI
директор Natalia de La Fournière

SAS ARTEFAKT AI
53 rue Felix Thomas
44000 NANTES
Siret : 823 773 387 00019
APE: 6201Z



ARTEFAKT AI

Siren: 823 773 387
R.C.S Nantes, France

23.06.2021

«Je valide»

Présidente ARTEFAKT AI
Natalia de La Fournière

ACTE DE MISE EN ŒUVRE

Cet acte est une confirmation de l'utilisation des résultats de la recherche Smorodin Andrii, qui ont été obtenus dans la thèse de doctorat sur «Méthodes d'apprentissage des réseaux de neurones basés sur la dynamique non linéaire ».

Dans la période du 02.01 .2020 au 26.02.2021 la société Artefakt AI a mené un ensemble d'études A l'aide d'un algorithme, qui a été développé par Smorodin Andrii, l'étudiant diplômé de l'Université d'Etat "Odessa Polytechnique" qui est en train de préparer une thèse de doctorat. Cet algorithme a été utilisé pour construire des réseaux de neurones de diverses architectures, qu'Artefakt AI utilise pour créer des odontogrammes dentaires à partir des radiographies dentaires panoramiques 2D. Les réseaux de neurones construits à l'aide d'une nouvelle méthode d'optimisation développée par l'étudiant diplômé Smorodin Andrii seront utilisés comme modules intégrés dans les logiciels de traitement des radiographies dentaires et dans les appareils d'acquisition des radiographies dentaires.

Grâce à l'utilisation du nouvel algorithme, il a été possible d'obtenir une accélération de 20 % du processus d'apprentissage pour l'algorithme de « stochastic gradient decent », ce qui a permis de réduire le temps d'apprentissage des réseaux de neurones et de réduire le coût des solutions clusterisées. Au cours du même temps d'entraînement, des réseaux de neurones avec des capacités prédictives plus élevées ont été obtenus.

ARTEFAKT AI

Présidente Natalia de La Fournière

SAS ARTEFAKT AI
53 rue Felix Thomas
44000 NANTES
Siret : 823 773 387 00019
APE: 6201Z

**ДОДАТОК Б. Довідка про використання результатів дисертаційної роботи у
науково-дослідницької діяльності**

Довідка

про використання результатів дисертаційної роботи
Смородіна Андрія Вячеславовича
«Методи навчання нейронних мереж на основі нелінійної
динаміки»

у науково-дослідницької діяльності Державного університету «Одеська
політехніка»

Довідка видана в тому, що у науково-дослідницькій діяльності **Державного університету «Одеська політехніка»** використані наступні наукові результати, отримані у дисертаційній роботі **Смородіна Андрія Вячеславовича**:

- програмні модулі, які розширили оптимізаційний блок бібліотеки PyTorch за рахунок удосконалених методів градієнтного спуску на основі положень нелінійної динаміки;
- методи прискорення часу навчання нейронної мережі для вирішення завдань сегментації та класифікації зубів на панорамних рентгенівських знімках відносно SGD алгоритму.

Дисертацію виконано згідно тематичних планів НДР ОНПУ за період 2017 - 2021 рр.

Смородін Андрій Вячеславович був молодшим науковим співробітником за держбюджетною темою № 716-145 – «Інноваційна інтелектуальна інформаційна технологія аналізу та синтезу енергоефективних гідроаеродинамічних елементів та систем» (№ ДР 0121О111720).



Керівник НДР

№ 0121О111720, д.т.н., професор

Г.А. Оборський

С.Г. Антошук

«04» 01 2021

**ДОДАТОК В. Довідка про впровадження результатів дисертаційної роботи у
навчальний процес**

ДОВІДКА

про впровадження результатів дисертаційної роботи
Смородіна Андрія Вячеславовича
«Методи навчання нейронних мереж на основі нелінійної
динаміки»
у навчальний процес Державного університету «Одеська політехніка»

Довідка видана у результаті того, що в програмах, навчально-методичних матеріалах та курсах лекцій з дисциплін, що вивчаються на кафедрі Інформаційних систем Інституту комп'ютерних систем бакалаврами спеціальності 122 «Комп'ютерні науки» – «Системи та методи штучного інтелекту» та магістрами спеціальності 122 «Комп'ютерні науки» «Обчислювальний інтелект та машинне навчання».

В навчальному процесі використовуються такі наукові результати, отримані у дисертації **Смородіна Андрія Вячеславовича**:

- дисципліна «Системи та методи штучного інтелекту»: ієрархічна нейромережева модель периферійних обчислень з використанням згорткових та рекурентних нейронних мереж – Тема «Згорткові нейронні мережі».
- дисципліна «Обчислювальний інтелект та машинне навчання»: методи компресії нейронних мереж на основі обчислення ймовірності виключення нейронів з урахуванням параметра надмірності – Тема «Методи проектування глибинних нейронних мереж».

Використані результати дисертації **Смородіна Андрія Вячеславовича** «Методи навчання нейронних мереж на основі нелінійної динаміки» свідчать про глибоке опрацювання наукової літератури за тематикою дисертації. Висновки та пропозиції відзначаються науковою новизною, можливістю реалізації для завдань автоматизованого аналізу мультисенсорних даних та зображень, тому істотно поліпшують матеріал означених дисциплін і сприяють підвищенню якості підготовки фахівців в галузі інтелектуального аналізу даних та особливо в напрямку нейромережевих технологій.



Г.А. Оборський

С.Г. Антошук

«04» 01 2022

ДОДАТОК Г. Список публікацій здобувача за темою дисертації

Статті у періодичних наукових виданнях інших держав, які входять до Європейського Союзу

1. Dmitrishin D., Smorodin A., Stokolos A. On a family of extremal polynomials. *Comptes Rendus Mathematique*. 2019. 357(7). P. 591-596. DOI:<https://doi.org/10.1016/j.crma.2019.06.010>. (Scopus, Q1).
2. Dmitrishin D., Smorodin A., Stokolos A., Tohaneanu M. Symmetrization of Suffridge polynomials and approximation of T-symmetric Koebe functions. *Journal of Mathematical Analysis and Applications*. 2021. 503(3). DOI:<https://doi.org/10.1016/j.jmaa.2021.125350>. (Scopus, Q1).
3. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. *Applied and Computational Harmonic Analysis*. 2021. 56. P. 283-305. DOI:<https://doi.org/10.1016/j.acha.2021.08.008>. (Scopus, Q1)

Статті у фахових наукових виданнях України

4. Smorodin A. Predictive control methods in tasks of searching saddle points. *Proceedings of Odessa Polytechnic University*. 2020. V. 3(62). P. 80-90. DOI:10.15276/opu.3.62.2020.10. URL:<https://pratsi.op.edu.ua/articles/show/20968>
5. Smorodin A. The use of control theory methods in training neural networks on the example of teeth recognition on panoramic x-ray images//Automation of Technological and Business Processes. 2021. V. 13(2). P. 36-40. URL:<http://dx.doi.org/10.15673/atbp.v13i2.2055>
6. Smorodin, A.V. Використання методів теорії управління при вивченні нейронних мереж на прикладі рукописного тексту. Прикладні аспекти інформаційних

технологій. V. 4(3). 2021. P. 243–249. URL:<http://aaait.ccs.od.ua/index.php/journal/article/view/119>

7. Dillies, J., Dmitrishin, D., Smorodin, A., Stokolos A. (2022). On the Koebe Quarter Theorem for Polynomials. Proceedings of the International Geometry Center, V. 14(3). 2022. P. 219-230. <https://doi.org/10.15673/tmgc.v14i3.2057>. (Scopus, Q4)

Опубліковані праці апробаційного характеру

8. Dmitrishin D., Smorodin A., Stokolos A. An extremal problem for polynomials. Harmonic Analysis and Partial Differential Equations. Canadian Mathematical Society. 2021.

9. Dmitrishin D., Smorodin A., Stokolos A. Extremal polynomials and geometric complex analysis. Algebraic and Geometric Methods of Analysis. International Online Conference. 2020.

10. Smorodin A. On question by Dimitrov. Analysis Seminar. Odessa. I. I. Mechnikov National University. 2018.

11. Смородин А. Методи прогнозного управління в задачах пошуку сидлових точок та навчання нейронних мереж. /Актуальні питання фізико-математичних та технічних наук: теоретичні та прикладні дослідження. 2021. Р. 13-23. II Міжнародна науково-практична інтернет-конференція. URL: https://openscilab.org/wp-content/uploads/2021/05/aktualni-pitannja-fiziko-matematichnih-ta-tehnichnih-nauk-teoretichni-ta-prikladni-doslidzhennja_2021_05_21.pdf

12. Смородин А. Методи прогнозного управління для задач оптимізації. Теоретико-практичні проблеми використання математичних методів та комп'ютерно-орієнтованих технологій в освіті та науці. 2021. III Всеукраїнська науково-практична

онлайн-конференція.

13. Dmitrishin D., Stokolos. A, Smorodin A. On a new family of extremal polynomials. The 35th annual Southeastern Analysis. Meeting, University of Alabama, 2019.

14. Dmitrishin D., Stokolos. A, Smorodin A. An extremal problem for polynomials. CMS 75th +1 Anniversary Summer Meeting, Montreal, 2021

15. Dmitrishin D., Skrinnik I., Smorodin A., Stokolos A. Dimitrov's question for the polynomials of degree 1,2,3,4,5,6. 2018. arXiv:1808.08636. URL: <https://arxiv.org/abs/1808.08636>.

16. Dmitrishin D., Smorodin A., Stokolos A. Estimating the Koebe radius for polynomials. 2018. arXiv: 1805.06927. URL: <https://arxiv.org/abs/1805.06927v1>

17. Dmitrishin D., Smorodin A., Stokolos. On C. Michel's hypothesis about the modulus of typically real polynomials. 2020. arXiv: 2005.12432. URL: <https://arxiv.org/abs/2005.12432>

18. Dillies J., Dmitrishin D., Smorodin A., Stokolos. On the Koebe Quarter Theorem for Polynomials. 2019. arXiv: 1904.11039. URL: <https://arxiv.org/abs/1904.11039>