

УДК 004.852

DOI <https://doi.org/10.32782/tnv-tech.2023.6.9>

НЕЙРОЕВОЛЮЦІЙНИЙ МЕТОД КОЛОКАЦІЇ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Ярош А. О. – аспірантка кафедри програмної інженерії
Запорізького національного університету
ORCID ID: 0009-0003-3495-9009

Кудін О. В. – кандидат фізико-математичних наук, доцент,
доцент кафедри програмної інженерії
Запорізького національного університету
ORCID ID: 0000-0002-5917-9127

Важливість розвитку наближених методів розв'язання диференціальних рівнянь визначається їхнім широким застосуванням у важливих галузях науки та техніки. Факт того, що багато фізичних та інженерних явищ можна математично описати диференціальними рівняннями, але часто важко знайти їхні аналітичні розв'язки, робить чисельні методи наближеного розв'язання критично важливими. Ці методи необхідні для комп'ютерного моделювання та симуляції складних технічних систем. Враховуючи широкий спектр різновидів диференціальних рівнянь, наближені методи стають універсальним інструментом, адаптованим для вирішення складних задач у різних галузях, та дозволяють краще враховувати вимоги сучасних обчислювальних технологій.

Застосування нейронних мереж для наближеного розв'язання диференціальних рівнянь представляє собою перспективний напрямок в галузі наукового моделювання. Нейронні мережі з додаванням фізичної інформації у вигляді складної функції втрат є інноваційним підходом, що об'єднує традиційні методи розв'язання фізичних задач із сучасними техніками глибокого навчання. У цьому підході, нейронна мережа, яка зазвичай використовується для апроксимації функцій, отримує на вхід не лише вхідні дані, але й фізичну інформацію про систему чи процес, яку вона моделює. Ця фізична інформація може бути включена у вигляді додаткових параметрів, обмежень чи рівнянь. Складна функція втрат враховує якість апроксимації нейронною мережею, а також фізичні принципи задачі. Це дозволяє нейронним мережам адаптуватися до фізичних обмежень і забезпечує наближене розв'язання задач, враховуючи важливі аспекти фізичної структури. В роботі досліджується можливість застосування генетичних алгоритмів для налаштування гіперпараметрів нейронних мереж, що апроксимують невідому функцію.

Ключові слова: чисельні методи, нейронні мережі, генетичний алгоритм, апроксимація.

Yarosh A. O., Kudin O. V. Neuroevolutionary collocation method for solving differential equations

The importance of the development of numerical methods for solving differential equations is determined by their wide application in important fields of science and technology. The fact that many physical and engineering phenomena can be mathematically described by differential equations, but it is often difficult to find their analytical solutions. This makes numerical methods of approximate solution crucial. These methods are necessary for computer modeling and simulation of complex technical systems. Taking into account the wide range of types of differential equations, approximate methods become a universal tool, adapted to solve complex problems in various fields, and allow better consideration of the requirements of modern computing technologies.

The use of neural networks for the approximate solution of differential equations is a promising direction in the field of scientific modeling. Neural networks with the addition of physical information in the form of a complex loss function are an innovative approach that combines traditional methods of solving physical problems with modern techniques of deep learning. In this approach, a neural network, which is typically used to approximate functions, receives as input not only input data but also physical information about the system or process it is modeling.

This physical information can be included as additional parameters, constraints, or equations. The complex loss function takes into account the quality of approximation by the neural network, as well as the physical principles of the problem. This allows neural networks to adapt to physical constraints and provides an approximate solution of problems, taking into account important aspects of the physical structure. The paper examines the possibility of applying genetic algorithms to adjust the hyperparameters of neural networks approximating an unknown function.

Key words: numerical methods, neural networks, genetic algorithms, approximation.

Вступ. Нейронні мережі набули широкого поширення останніми роками в таких сферах як автоматичний переклад, комп'ютерний зір тощо. Однак, на цьому ефективні застосування нейромережових моделей не обмежуються. Використання нейронних мереж у інженерних задачах включає прогнозування та оптимізацію, керування та автоматизацію, обробку сигналів та візуальне розпізнавання, проектування та імітацію, а також дослідження та розробку нових технологій. Так, наприклад, робота [1] містить розв'язання задачі ідентифікації параметрів авіаційного двигуна ТВ3-117 у бортових умовах із застосуванням нейронних мереж. У загальному випадку, в інженерних застосуваннях, зазвичай, використовуються математичні моделі, які описуються диференціальними рівняннями як лінійними, так і нелінійними [2].

Наближені методи розв'язання крайових задач для диференціальних рівнянь та систем з частинними похідними, такі як метод Рітца, Гальоркіна, колокації або скінченних елементів добре обґрунтовані та мають велику кількість застосувань для різних задач [3].

Постановка задачі. Останніми роками у наукових обчисленнях сформувався напрям “scientific machine learning” (SciML) або “Physics-informed machine learning” (PIML) [4, 5], особливістю якого є застосування методів машинного навчання у моделюванні вимогливих до ресурсів наукових задач. Основою для таких методів є теореми про збіжність апроксимації нейронними мережами [6]. Ідея цих підходів полягає у заміні невідомої функції та її похідних нейронною мережею, далі виконуються операції, специфічні для кожного методу (Рітца, Гальоркіна або колокації) [7, 8]. Результатом є нейронна мережа з параметрами, що відповідають диференціальному рівнянню та крайовим умовам. Такі варіанти класичних методів отримують назву глибинних [9–17].

Застосування нейронної архітектури додає чисельним методам такі переваги [5]:

- штучні нейронні мережі дозволяють апроксимувати нелінійні залежності довільної складності, налаштування параметрів мережі відбувається під час навчання;
- такі методи є загальними і можуть бути застосованими до звичайних диференціальних рівнянь та рівнянь у частинних похідних;
- ефективно працюють на задачах високої розмірності;
- глибинні методи можуть бути ефективно реалізовані на паралельних архітектурах.

До недоліків можна віднести:

- необхідність налаштування гіперпараметрів нейромереж, що може бути обчислювально складною задачею;
- недостатня точність глибинних методів у порівнянні з класичними;

Актуальною задачею є розвиток обчислювальних методів розв'язання диференціальних рівнянь та їх систем у напрямі розширення застосування нейромереж для розв'язання фізичних та інженерних задач, сформульованих у вигляді диференціальних рівнянь.

Мета дослідження. Метою даної роботи є розробка варіанта нейромережевого методу колокації з використанням генетичного алгоритму для оптимізації гіперпараметрів нейромережі.

Об'єктом дослідження є процес розв'язання диференціальних рівнянь засобами нейромережевих методів. Гіпотеза дослідження полягає в тому, що використання еволюційних методів пошуку гіперпараметрів нейромереж, які є апроксимаціями шуканих функцій, може збільшити точність розв'язку. Припускається, що використання нейромережевого варіанта методу колокації з генетичною оптимізацією мережі підвищить точність алгоритму на тестовій задачі. Основними критеріями є точність побудованого розв'язку та збіжність генетичного алгоритму.

Аналіз останніх досліджень і публікацій. Статтю [7] присвячено розробці загального методу розв'язання звичайних диференціальних рівнянь та рівнянь у частинних похідних, який використовує нейронні мережі для апроксимації невідомої функції. Використовується мережа прямого поширення сигналу, параметри якої налаштовуються при мінімізації відповідної функції втрат. В свою чергу, функція втрат складається з двох частин. Перший член відповідає початковим або граничним умовам задачі. Другий член задає нейронну мережу, яка повинна задовольняти диференціальному рівнянню. Особливість цього методу полягає в тому, що розв'язок представляється у замкнутій диференційованій формі, яку можна використовувати у подальших обчисленнях. В той час як традиційні методи пропонують дискретний розв'язок (метод Рунге-Кутта, послідовних наближень тощо). Демонструється збіжність запропонованого методу з точними розв'язками модельних задач.

В роботі [8] розробляється підхід до навчання нейронних мереж на основі даних, що описують деякий фізичний процес. Автори пропонують використовувати апріорні знання про відповідні фізичні закони та гіпотетичні залежності як регуляризатори функції втрат нейромережі. В залежності від характеристик наявних даних, розроблено два типи моделей: з неперервною та дискретною часовою шкалою. Перший тип може використовуватись для апроксимації просторово-часових функцій. Моделі другого типу передбачають ітераційний процес з кроком за часом. В роботі розглянуто параметричні та нелінійні диференціальні рівняння в частинних похідних.

Стаття [9] присвячено розробці нейромережевого варіанта методу Гальоркіна розв'язання багатовимірних параболічних диференціальних рівнянь. Цей варіант методу в цілому відповідає класичному підходу та має такі основні етапи: невідома функція замінюється нейронною мережею, із застосуванням методу автоматичного диференціювання обчислюються необхідні похідні; формується цільова функція, яка є комбінацією квадратичних відхилень значень рівняння та граничних умов; генерується випадкова множина пробних точок з області визначення шуканої функції та граничних умов; обчислюється значення нев'язок цільової функції у випадкових точках; застосовується крок градієнтного спуску до значень параметрів нейронної мережі, причому параметр швидкості навчання зменшується зі зростанням кількості ітерацій алгоритму. Отже, нейромережевий варіант методу Гальоркіна замінює базисні функції на нейронну мережу. Під час навчання мережі стохастичним градієнтним спуском налаштовуються параметри нейромережі з урахуванням диференціального рівняння та крайових умов.

Нейромережевий варіант методу Рітца пропонується в роботі [10]. Основна ідея цього підходу також схожа на попередні з врахуванням того, що цей метод застосовується для варіаційних задач. Функції апроксимації замінюються на

нейромережу з параметрами, які налаштовуються під час навчання методом градієнтного спуску.

Робота [11] присвячена адаптації нейромереж до метода колокації на прикладі розв'язання задачі згину тонких квадратних та круглих пластин. Результати обчислювальних експериментів демонструють узгодженість прогнозованої деформації пластини з точним розв'язком. Зазначається, що збільшення кількості шарів нейронної мережі прямого поширення сигналу та кількості нейронів в них, прогнозоване значення наближається до точного. При цьому, використовувались випадкові точки колокації, середня квадратична похибка для оцінки функції втрат та варіант метода градієнтного спуску з адаптивною швидкістю навчання.

Бібліотека розв'язання диференціальних рівнянь DeepXDE, яка є Python реалізацією підходу на основі нейронних мереж з додатковою фізичною інформацією розглядається в статті [12]. Пропонується метод адаптивного уточнення та основні залишків. Результати моделювання порівнюються з методом скінченних елементів. Розглядається задачі апроксимації заданої функції нейронною мережею, розв'язання звичайних диференціальних рівнянь та рівнянь у частинних похідних, а також обернена проблема диференціальних рівнянь в частинних похідних. Бібліотека широко застосовується у наукових дослідженнях, зокрема в [13] фізично-інформовані нейронні мережі застосовуються в задачах оптимізації.

В [14] мережі з додатковою фізичною інформацією застосовуються для розробки системи неперервного моніторингу стану механічної системи на основі даних. Розв'язується задача прогину балки Ейлера-Бернуллі. Розглядається розподілене поперечне та точкове навантаження. Результати порівнюються з аналітичним та скінченно-елементним розв'язками, продемонстрована задовільна збіжність нейромережевого метода.

Роботу [15] присвячено використанню функціоналу першого порядку методу найменших квадратів у якості функції втрат нейромережі прямого поширення сигналу. Метод використовується для розв'язання еліптичних диференціальних рівнянь.

В статті [16] представлено архітектуру нейромережі динамічного глибокого навчання на основі методу скінченних елементів для розв'язання лінійних параметричних диференціальних рівнянь з частинними похідними. Під час уточнення сітки зв'язки між нейронами в архітектурі мережі імітують графік зв'язності скінченних елементів. Розглянуто декілька функцій втрат. Метод реалізовано для просторової області 1D.

Нейронні мережі радіально базисних функцій (РБФ мережі) з фізичною інформацією розробляються в роботі [17]. На відміну від глибоких нейронних мереж, радіальна базисна мережа містить тільки один прихований шар і відповідно радіальні базисні функції активації. Продемонстровано, що даний тип мереж з використанням методів градієнтного спуску є збіжним. Чисельні приклади показали, що РБФ мережі є більш ефективним у розв'язанні нелінійних диференціальних рівнянь в частинних похідних, ніж глибинні нейромережі з фізичною інформацією.

Можна зробити висновок, що нейронні мережі різних архітектур з успіхом застосовуються для розв'язання звичайних диференціальних рівнянь та рівнянь у частинних похідних. Особливістю є те, що шукані функції апроксимуються нейромережами, частіше, прямого поширення сигналу. Рівняння та крайові умови, зазвичай, входять у функції втрат як додаткові регуляризатори. Для оптимізації часто використовуються різні адаптивні варіанти метода градієнтного спуску.

Обчислювальні експерименти демонструють задовільні результати для лінійних рівнянь, навіть, коли невідома функція має велику розмірність. Менше висвітлено питання розв'язання нелінійних диференціальних рівнянь засобами глибинних методів. В розглянутих роботах не приділено уваги питанню пошуку оптимальних наборів гіперпараметрів, які задають структуру нейромережі, а отже, впливають на точність розв'язання. В той же час, налаштування гіперпараметрів нейромережі є критичним кроком у розв'язанні практичних завдань за допомогою глибокого навчання. Гіперпараметри включають різноманітні аспекти, такі як архітектурні рішення (кількість шарів, нейронів, функції активації), розміри пакетів для тренування та інші.

Виклад основного матеріалу дослідження. Ідея нейромережевого метода колокації для розв'язання диференціальних рівнянь полягає у заміні шуканої функції на нейромережу. Без обмеження загальності, розглянемо приклад роботи алгоритму для диференціального рівняння другого порядку [3].

$$L[y] \equiv \frac{d^2 y}{dx^2} + g(x) \frac{dy}{dx} + h(x)y = f_0(x),$$

крайові умови мають вигляд

$$\Gamma_a \equiv \pm_1 \frac{dy}{dx}(a) + \pm_2 y(a) = A,$$

$$\Gamma_b \equiv \beta_1 \frac{dy}{dx}(b) + \beta_2 y(b) = B,$$

де \pm_1, \pm_2 – деякі константи, причому $|\alpha_1| + |\alpha_2| \neq 0, |\beta_1| + |\beta_2| \neq 0$.

Отже, згідно класичного методу колокації, наближений розв'язок представляється у вигляді лінійної комбінації незалежних функцій $u_0(x), u_1(x), \dots, u_n(x)$. Функції обираються так, щоб вони задовольняли крайові умови Γ_a, Γ_b .

Наближений розв'язок визначається так:

$$\bar{y}(x) = u_0(x) + \sum_{i=1}^n c_i u_i(x).$$

Нев'язки в цьому випадку мають вигляд

$$R(x, c_1, \dots, c_n) \equiv L[\bar{y}] - f_0(x) = L[u_0] - f_0(x) + \sum_{i=1}^n c_i L[u_i].$$

Константи c_i визначаються з системи лінійних алгебраїчних рівнянь

$$R(x_1, c_1, \dots, c_n) = 0,$$

$$R(x_2, c_1, \dots, c_n) = 0,$$

$$R(x_n, c_1, \dots, c_n) = 0.$$

Де x_1, x_2, \dots, x_n – деякий набір точок з відрізка $[a, b]$, так звані, точки колокації.

Особливість нейромережевого методу в тому, що невідома функція замінюється на нейронну мережу $W(x, \Theta)$, де Θ – гіперпараметри мережі [11]:

$$G(W(x, \Theta)) \equiv \frac{d^2 W(x, \Theta)}{dx^2} + g(x) \frac{dW(x, \Theta)}{dx} + h(x)W(x, \Theta) = f_0(x),$$

$$\Gamma_a \equiv \alpha_1 \frac{dW(x, \Theta)}{dx}(a) + \alpha_2 W(a, \Theta) = A,$$

$$\Gamma_b \equiv \beta_1 \frac{dW(x, \Theta)}{dx}(b) + \beta_2 W(b, \Theta) = B.$$

Функція втрат визначається як середнє квадратичне відхилення (Mean Squared Error, MSE) результатів нейромережі від правої частини відповідно диференціального рівняння та крайових умов:

$$MSE = MSE_G + MSE_{G_a} + MSE_{G_b}.$$

Така функція втрат і є поширеним способом інтегрувати в нейромережу додаткову фізичну інформацію.

Для автоматизації пошуку оптимальних нейронних мереж, широко використовуються еволюційні алгоритми [18], зокрема, класичні генетичні алгоритми Дж. Г. Голланда [19].

Процес використання генетичного алгоритму може включати такі етапи [19].

1. Визначення простору гіперпараметрів. Визначаються гіперпараметри нейромережі, які підлягають оптимізації. Це може включати, наприклад, кількість шарів та нейронів в нейромережі, швидкість навчання, розміри пакетів для тренування, тип функцій активації тощо.

2. Створення початкової популяції. Генерується випадковий набір параметрів для нейромережі, який складає початкову популяцію. Кожен варіант параметрів представляє індивіда в популяції.

3. Оцінка пристосованості. Кожен індивід у популяції оцінюється за якістю його моделі на валідаційному наборі даних. Це може включати оцінку точності, витрати функції, чи інші метрики відповідно до конкретної задачі.

4. Вибір батьківських розв'язків. Вибираються індивіди для розмноження (створення нащадків) на основі їхньої пристосованості. Індивіди з вищою пристосованістю мають більше шансів бути обраними для розмноження.

5. Створення нащадків. Застосовуються генетичні оператори, такі як кросовер (комбінування частин гіперпараметрів батьків), мутація (випадкові зміни гіперпараметрів) та інші, для створення нової популяції.

6. Оцінка пристосованості нової популяції. Нова популяція оцінюється на валідаційному наборі даних, і цей процес повторюється протягом кількох поколінь.

7. Зупинка. Процес триває до досягнення задовільного рівня пристосованості або до вичерпання кількості поколінь.

Отже, основні оператори генетичного алгоритму: оператор відбору, кросовер та оператор мутації. Оператор відбору визначає, які особини обираються для подальших операцій, таких як кросовер чи мутація. Кросовер використовується для обміну генетичною інформацією між батьками, створюючи елементи наступної популяції. Мутація вносить випадкові зміни в генетичні коди потенційних розв'язків.

Генетичні алгоритми є ефективним ітеративним методом для глобальної оптимізації, і в їхньому використанні для налаштування гіперпараметрів нейромереж полягає можливість обійти простір гіперпараметрів та знаходити оптимальні комбінації для досягнення кращих результатів на валідаційних даних.

Формально, початковий стан алгоритму *Gen* генетичного пошуку можна описати у вигляді такої функції [19]:

$$Gen = Gen(P_0, N, L, f, \Omega, \Psi, \vartheta, T),$$

де $P_0 = \{H_1^0, H_2^0, \dots, H_N^0\}$ – початкова популяція – певна початкова комбінація гіперпараметрів, поданих у вигляді хромосом; $H_j^0 = \{h_{1j}^0, h_{2j}^0, \dots, h_{L_j}^0\}$ – j -та хромосома популяції, набір значень гіперпараметрів, поданих у вигляді генів; h_{ij}^0 – i -ий ген j -ої хромосоми популяції P_0 – тобто, значення i -го оптимізованого параметру

задачі, що входить в j -те рішення; N – кількість хромосом в популяції; L – довжина хромосом, кількість генів; f – цільова функція; \odot – оператор відбору; Ψ – оператор схрещування; Θ – оператор мутації; T – критерії зупинення.

В термінах цього підходу, хромосома є закодованим варіантом нейронної мережі певної структури, тобто гени хромосоми відповідають певним гіперпараметрам мережі: 1) тип ініціалізатора ядра; 2) тип ініціалізатора зсуву; 3) кількість шарів; 4) кількість нейронів (однакова для кожного шару); 5) тип оптимізатора; 6) розмір пакету навчання; 7) тип функції активації; 8) швидкість навчання.

Всі гени кодуються цілими числами. Тип кросовера – одноточковий. Тип мутації – випадковий. Ймовірність мутації – 70%, 60% генів піддаються мутації.

Приклад. Розв'яжемо рівняння $\frac{dy}{dx} = y(x)x + y(x)^2$, $y(1) = 1$.

Точний розв'язок в спеціальних функціях має такий вигляд (<https://cutt.ly/tZXpJHM>):

$$y = \frac{e^{\frac{x^2}{2}}}{-\sqrt{2\pi} \operatorname{erfi}\left(\frac{x}{\sqrt{2}}\right) + \sqrt{2\pi} \operatorname{erfi}\left(\frac{1}{\sqrt{2}}\right) + 2\sqrt{e}}$$

де $\operatorname{erfi}(\)$ – функція помилок Гауса.

Збіжність значень генів подаються на рисунку 1.

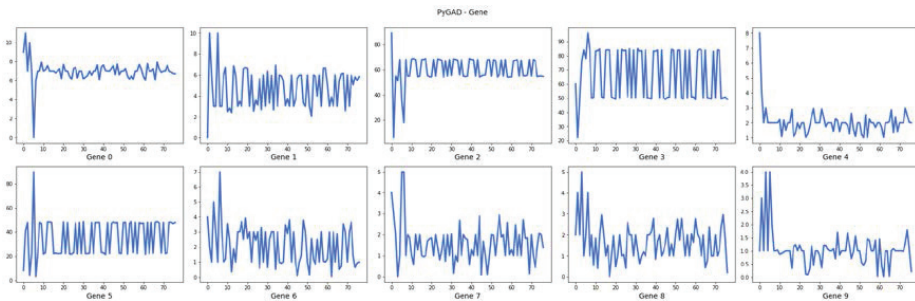


Рис. 1. Значення генів за генераціями

З рисунку 1 можна побачити збіжність значень генів при зростанні номера популяції, тобто структура нейронної мережі є відносно стійкою від генерації до генерації.

Чисельний розв'язок в десяти точках наведено в таблиці 1.

Таблиця 1

Результати чисельного експерименту

| Точка колокації | Точний розв'язок | Наближений розв'язок |
|-----------------|------------------|----------------------|
| 0 | 0.35166 | 0.37413 |
| 0.1 | 0.36632 | 0.38202 |
| 0.2 | 0.38610 | 0.40259 |
| 0.3 | 0.41197 | 0.43815 |

Продовження табл. 1

| | | |
|-----|---------|---------|
| 0.4 | 0.44529 | 0.48927 |
| 0.5 | 0.48800 | 0.55465 |
| 0.6 | 0.54280 | 0.63163 |
| 0.7 | 0.61364 | 0.71712 |
| 0.8 | 0.70640 | 0.80831 |
| 0.9 | 0.83024 | 0.90308 |
| 1.0 | 1.00000 | 1.00000 |

Порівняння наближеного та точного розв'язку наведено на рисунку 2.

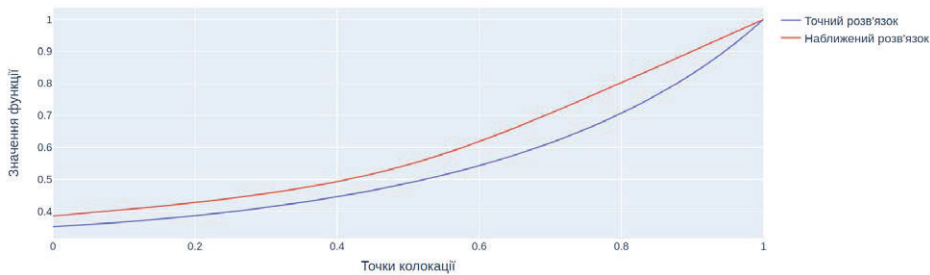


Рис. 2. Точний та наближений розв'язок

Висновки. Отже, в роботі запропоновано розширення алгоритму [11] з використанням генетичного алгоритму. Результати свідчать про збіжність алгоритму, відносна точність наближення в даному обчислювальному експерименті 10%. Програмну реалізацію наведено у Google Colab ноутбучі (<https://cutt.ly/DwXUHQcSJ>).

Перспективи подальших досліджень пов'язані з розвитком нейроеволюційних методів та розширенням на більш загальну область інженерних задач, наприклад, будівельну механіку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Vladov S., Shmelov Yu., Kotliarov K., Hrybanova S., Husarova O., Derevyanko I., Chyzhova L. Onboard parameter identification method of the TV3-117 aircraft engine of the neural network technologies. Вісник КрНУ імені Михайла Остроградського. Випуск 5/2019 (118), 2019. Р. 90-96.
2. Edwards C.H., Penney D.E., Calvis D.T. Differential Equations and Boundary Value Problems: Computing and Modeling. Boston: Pearson, 2014. 797p.
3. Pinder G.F. Numerical Methods for Solving Partial Differential Equations. John Wiley & Sons, Inc., 2018.
4. Karniadakis G.E., Kevrekidis I.G., L.Lu, P. Perdikaris, Wang S., Yang L., Physics-informed machine learning, Nat Rev Phys, vol. 3, no. 6, pp. 422–440, 2021, doi: 10.1038/s42254-021-00314-5.
5. Willard J., Jia X., Xu S., Steinbach M., Kumar V. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. ACM Comput. Surv., 2022, <https://doi.org/10.1145/3514228>
6. Cybenko G.V. Approximation by Superpositions of a Sigmoidal function, Mathematics of Control, Signals and Systems, , 1989, vol. 2 no. 4 pp. 303-314

7. Lagaris I.E., Likas A., Fotiadis D.I. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *arXiv:physics/9705023v1*, 1997, <https://doi.org/10.1109/72.712178>
8. Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 2019, 686–707.
9. Sirignano J., Spiliopoulos K. DGM: A deep learning algorithm for solving partial differential equations. *arXiv:1708.07469v5*, 2018.
10. Weinan E, Bing Yu. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Commun. Math. Stat.*, 2018, 6:1–12. <https://doi.org/10.1007/s40304-018-0127-z>
11. Hongwei Guo, Timon Rabczuk, and Xiaoying Zhuang. A Deep Collocation Method for the Bending Analysis of Kirchhoff Plate. *arXiv:2102.02617v1*, 2021.
12. Lu Lu, Xuhui Meng, Zhiping Mao, George Em Karniadakis. DEEPXDE: A Deep Learning Library For Solving Differential Equations. *arXiv:1907.04502v2*, 2020.
13. Seo J. Solving real-world optimization tasks using physics-informed neural computing. *Scientific Reports*, 14(1), 202, 2024.
14. Radbakhsh S.H., Zandi K., Nikbakht M.. Physics-informed neural network for analyzing elastic beam behavior. *Structural Health Monitoring*, 2023.
15. Cai Z., Chen J., Liu M., Liu X., Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic PDEs, *J. Comput. Phys.* 420, 2020, 109707, <http://dx.doi.org/10.1016/j.jcp.2020.109707>.
16. Uriarte C., Pardo D., Omella A.J. A Finite Element based Deep Learning solver for parametric PDEs. *Comput. Methods Appl. Mech. Engrg.* 391, 2022, 114562, <https://doi.org/10.1016/j.cma.2021.114562>
17. Bai J., Liu G.-R., Gupta A., Alzubaidi L., Feng X.-Q., Gu Y. Physics-informed radial basis network (PIRBN): A local approximation neural network for solving nonlinear PDEs, *Computer Methods in Applied Mechanics and Engineering*, 415, 2023. ISSN 0045-7825, <https://doi.org/10.1016/j.cma.2023.116290>.
18. Galvan E., Mooney P. Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges, 2020. <https://doi.org/10.48550/arXiv.2006.05415>
19. Holland J. *Adaptation in natural and artificial systems*. Ann Arbor : University of Michigan Press, 1975. 183 p.

REFERENCES:

1. Vladov S., Shmelov Yu., Kotliarov K., Hrybanova S., Husarova O., Derevyanko I., Chyzhova L. (2019) Onboard parameter identification method of the TV3-117 aircraft engine of the neural network technologies. *Visnyk KrNU imeni Mykhaila Ostrohradskoho*. Issue 5 (118). P. 90-96.
2. Edwards C.H., Penney D.E., Calvis D.T. (2014) *Differential Equations and Boundary Value Problems: Computing and Modeling*. Boston: Pearson. 797p.
3. Pinder G.F (2018) *Numerical Methods for Solving Partial Differential Equations*. John Wiley & Sons, Inc.
4. Karniadakis G.E., Kevrekidis I.G., Lu L., Perdikaris P., Wang S., Yang L. (2021) Physics-informed machine learning, *Nat Rev Phys*, vol. 3, no. 6, pp. 422–440, doi: 10.1038/s42254-021-00314-5.
5. Willard J., Jia X., Xu S., Steinbach M., Kumar V. (2022) Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *ACM Comput. Surv.*, <https://doi.org/10.1145/3514228>
6. Cybenko G.V. (1989) Approximation by Superpositions of a Sigmoidal function, *Mathematics of Control, Signals and Systems*, vol. 2 no. 4 pp. 303-314
7. Lagaris I.E., Likas A., Fotiadis D.I. (1997) Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *arXiv:physics/9705023v1*, <https://doi.org/10.1109/72.712178>

8. Raissi M., Perdikaris P., Karniadakis G.E. (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 686–707.
 9. Sirignano J., Spiliopoulos K. (2018) DGM: A deep learning algorithm for solving partial differential equations. arXiv:1708.07469v5
 10. Weinan E, Bing Yu. (2018) The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Commun. Math. Stat.*, 6:1–12. <https://doi.org/10.1007/s40304-018-0127-z>
 11. Hongwei Guo, Timon Rabczuk, Xiaoying Zhuang. (2021) A Deep Collocation Method for the Bending Analysis of Kirchhoff Plate. arXiv:2102.02617v1
 12. Lu Lu, Xuhui Meng, Zhiping Mao, George Em Karniadakis. (2020) DEEPXDE: A Deep Learning Library For Solving Differential Equations. arXiv:1907.04502v2
 13. Seo J. (2024) Solving real-world optimization tasks using physics-informed neural computing. *Scientific Reports*, 14(1), 202.
 14. Radbakhsh S.H., Zandi K., Nikbakht M. (2023) Physics-informed neural network for analyzing elastic beam behavior. *Structural Health Monitoring*.
 15. Cai Z., Chen J., Liu M., Liu X. (2020) Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic PDEs, *J. Comput. Phys.* 420, 109707, <http://dx.doi.org/10.1016/j.jcp.2020.109707>.
 16. Uriarte C., Pardo D., Omella A.J. (2022) A Finite Element based Deep Learning solver for parametric PDEs. *Comput. Methods Appl. Mech. Engrg.* 391, 114562, <https://doi.org/10.1016/j.cma.2021.114562>
 17. Bai J., Liu G.-R., Gupta A., Alzubaidi L., Feng X.-Q., Gu Y. (2023) Physics-informed radial basis network (PIRBN): A local approximation neural network for solving nonlinear PDEs, *Computer Methods in Applied Mechanics and Engineering*, 415. ISSN 0045-7825, <https://doi.org/10.1016/j.cma.2023.116290>.
 18. Galvan E., Mooney P. (2020) Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges. <https://doi.org/10.48550/arXiv.2006.05415>
 19. Holland J. (1975) *Adaptation in natural and artificial systems*. Ann Arbor : University of Michigan Press. 183 p.
-