

ПРОЄКТУВАННЯ БІБЛІОТЕКИ НЕЙРОМЕРЕЖЕВИХ МЕТОДІВ РОЗВ'ЯЗАННЯ КРАЙОВИХ ЗАДАЧ

Ярош А.О.

Аспірантка

Запорізький національний університет, Україна

Кудін О.В.

канд. фіз.-мат. наук, доцент, професор кафедри програмної інженерії

Запорізький національний університет, Україна

Крайові задачі в науці та техніці відіграють важливу роль, оскільки вони дозволяють математично описувати різноманітні фізичні явища і процеси з використанням диференціальних рівнянь та крайових умов. Методи розв'язання крайових задач є ключовими для розробки нових технологій, оптимізації дизайну та прогнозування поведінки систем. Наприклад, такі методи дозволяють оптимізувати конструкції пристроїв, розробляти нові матеріали з певними властивостями, прогнозувати тепло- та гідродинамічні процеси, а також покращувати енергоефективність технічних систем.

Класичні наближені методи розв'язання крайових задач, такі як метод Гальоркіна, метод Рітца, метод колокації та метод кінцевих різниць, мають свої особливості та обмеження. Загалом, ці методи базуються на апроксимації функцій, що описують розв'язок задачі, за допомогою скінченної кількості параметрів або базових функцій. Наприклад, у методі Гальоркіна використовується базис ортогональних функцій для апроксимації розв'язку, у методі Рітца - набір тестових функцій, у методі колокації - точки, в яких точно виконується диференціальне рівняння, а метод кінцевих різниць розбиває область на сітку і апроксимує похідні.

Важливість розробки нових наближених методів полягає в тому, що сучасні технології і завдання можуть вимагати більш точних, ефективних або складних підходів до розв'язання крайових задач. Нові методи можуть включати в себе застосування інтерполяції з використанням нейронних мереж (наприклад, нейронні мережі з фізичною інформацією).

Метод нейронних мереж з фізичною інформацією (Physics-Informed Neural Networks, PINN) [1] є потужним інструментом в галузі наукових обчислень, який поєднує в собі переваги нейронних мереж з фізичними обмеженнями задач. Основна особливість цього методу полягає в тому, що він дозволяє інтегрувати фізичні закони або обмеження в оптимізаційні задачі з використанням нейронних мереж.

У методі PINN нейронна мережа використовується для апроксимації функції, що описує фізичний процес, одночасно з урахуванням фізичних законів у функції втрат мережі. Це дозволяє ефективно вирішувати задачі інверсійного моделювання та розв'язування різних диференціальних рівнянь, використовуючи лише дані спостереження.

Однією з основних переваг PINN є можливість працювати з обмеженими або неповними даними, які зазвичай важко обробляти за допомогою класичних методів. Крім того, метод може автоматично враховувати фізичні закони у вигляді обмежень під час навчання, що дозволяє покращити точність і стійкість моделі.

Загалом, метод PINN відкриває нові можливості для вирішення складних задач на перетині штучного інтелекту та фізичного моделювання, забезпечуючи ефективний інструмент для аналізу, прогнозування та оптимізації різноманітних фізичних процесів [1].

Бібліотека DeepXDE (Deep eXtreme Learning for Differential Equations) [2] використовується для розв'язання диференціальних рівнянь за допомогою глибокого навчання. Вона поєднує сучасні методи глибокого навчання з чисельними методами для ефективного вирішення різних видів диференціальних рівнянь, включаючи звичайні диференціальні рівняння, часткові диференціальні рівняння і інші. DeepXDE підтримує різноманітні типи задач, включаючи теплопровідність, гідродинаміку, механіку твердого тіла тощо, і може працювати з різними крайовими умовами. Крім того, бібліотека має зручний інтерфейс, що спрощує визначення моделей, навчання та отримання результатів, що робить її доступною для використання дослідниками і інженерами з різним рівнем досвіду у глибокому навчанні. Використання бібліотеки DeepXDE на даний час досить поширене в наукових публікаціях [3, 4]. Однак, певним недоліком бібліотеки є те, що ефективно її використання потребує знання мов програмування та дотримання певного внутрішнього формату запису задачі, приклад використання бібліотеки можна побачити в офіційній документації [5].

В даній роботі запропоновано проєкт об'єктно-орієнтованої бібліотеки AutoPINN, яка містить реалізацію PINN мереж з можливістю використання генетичних алгоритмів для налаштування гіперпараметрів мережі. Досить поширеним підходом у машинному навчанні є використання AutoML [6]. При цьому, користувачу достатньо виконати постановку задачі, а подальший аналіз виконується в автоматичному режимі з генерацією детальних звітів. Застосування подібного до AutoML методу взаємодії з користувачем у бібліотеці нейромережевих методів розв'язання крайових задач зводиться до математично коректної постановки задачі у LaTeX форматі. Потім користувач може вказати гіперпараметри нейромережі або обрати автоматичний підбір генетичним алгоритмом, після чого сформується результат.

На рисунку 1 наведено концептуальну UML діаграму основних класів бібліотеки. Так, клас ANN містить поля та методи для визначення архітектури нейромережі: кількості шарів (`n_layers`), кількості нейронів в шарах (список `n_units`), список активацій в шарах (`activations`) та об'єкт PyTorch, який містить власне нейромережу. Метод `forward()` – програмує прямий хід обчислень мережі.

Клас Net містить PINN мережу, яка є наближеним методом розв'язання крайової задачі. Поле `model` містить екземпляр класу ANN. Набори даних `X_train` та `y_train` описують поведінку в області визначення та області значень невідомої функції. Ці набори кодуєть початкові та граничні умови. Поля `optimizer` та `criterion` містять відповідно оптимізатор та метрику похибки бібліотеки PyTorch.

Метод `loss_func()` відповідає за формування функції втрат мережі, яка містить похибку на крайових умовах та похибку диференціального рівняння. В цьому методі відбувається диференціювання мережі. Метод `train()` виконує навчання PINN мережі.

Клас GenANN використовується для налаштування гіперпараметрів нейромережі генетичним алгоритмом. Поля класу:

- `populationSize`: розмір популяції генетичного алгоритму (кількість особин у популяції);
- `mutationRate`: ймовірність мутації при генерації нових особин;
- `crossoverRate`: ймовірність перехресного схрещування між батьками для створення нащадків;

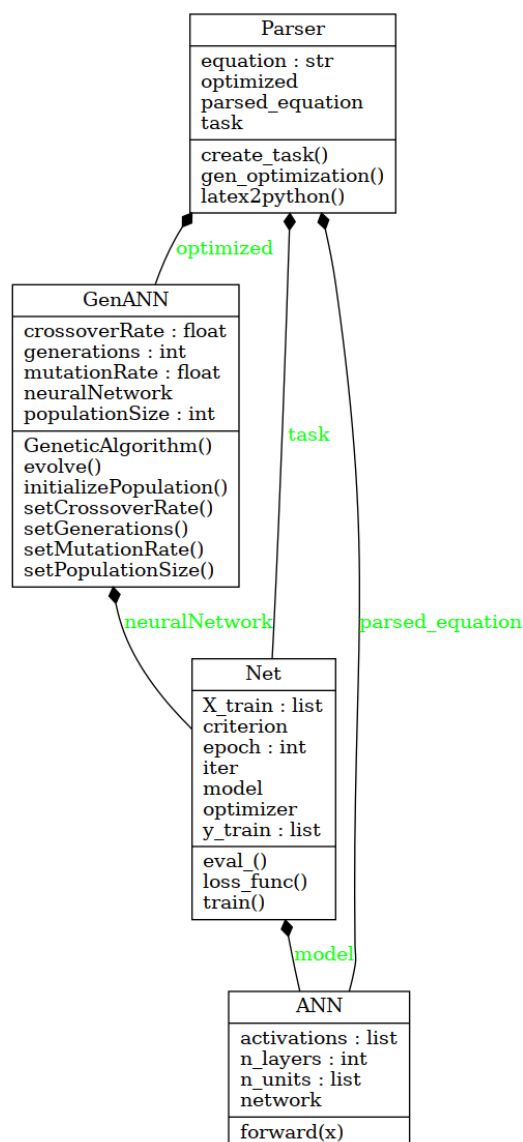


Рис. 1. Діаграма класів бібліотеки AutoPINN

- generations: кількість поколінь (ітерацій) для виконання генетичного алгоритму;

- neuralNetwork: об'єкт нейромережі, який буде налаштовуватись за допомогою генетичного алгоритму;

Методи класу GeneticAlgorithm:

- setPopulationSize: метод для встановлення розміру популяції;

- setMutationRate: метод для встановлення ймовірності мутації;

- setCrossoverRate: метод для встановлення ймовірності перехресного схрещування;

- setGenerations: метод для встановлення кількості поколінь;

- setNeuralNetwork: метод для встановлення об'єкту нейромережі;

- initializePopulation: метод для створення початкової популяції геномів (особин);

- evolve(): метод для запуску генетичного алгоритму: розмноження, селекції, перехресного схрещування та мутації популяції.

Клас Parser містить методи перетворення математичного запису крайової задачі у LaTeX форматі. Метод latex2python() перетворює текстовий запис математичної

задачі у послідовність викликів функцій Python. Метод `create_task()` формує необхідні об'єкти класів Net та ANN, тобто формує задачу для розв'язку. Метод `gen_optimization()` створює екземпляр класу GenANN та виконує генетичну оптимізацію мережі.

Запропонована бібліотека дозволить розширити практику використання нейронних мереж як обчислювальних методів розв'язання крайових задач. Репозиторій прототипу бібліотеки знаходиться за посиланням <https://github.com/avk256/AutoPINN>.

Список використаних джерел:

1. Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural net-works: A deep learning framework for solving forward and inverse prob-blems involving nonlinear partial differential equations. *Journal of Computa-tional Physics* 378, 2019, 686–707..
2. Lu L., Meng X., Mao Z. and Karniadakis G. Em. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, volume 63, 1,pp. 208-228, 2021, DOI: 10.1137/19M1274067
3. Seo J. Solving real-world optimization tasks using physics-informed neural computing. *Scientific Reports*, 14(1), 202, 2024.
4. Wu J., Wu Y., Zhang G., Zhang Y. Variable linear transformation improved physics-informed neural networks to solve thin-layer flow problems. *Journal of Computational Physics*, 112761, 2024.
5. Офіційна документація бібліотеки DeepXDE. URL: https://deepxde.readthedocs.io/en/latest/demos/pinn_forward/ode.2nd.html (дата звернення 20.04.2024)
6. Ferreira L., Pilastrri A., Martins C. M., Pires P. M. and Cortez P., A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost. 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534091.