**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота**

**на здобуття ступеня бакалавра**

за спеціальністю 121 Інженерія програмного забезпечення

на тему:

**Адаптація нейромережевих та евристичних алгоритмів для побудови моделей динамічних процесів**

|  |  |  |
| --- | --- | --- |
| Виконав студент 4-го курсу  Максим ГАВРИЛЮК | | \_\_\_\_\_\_\_\_\_ |
|  | |  |
| Науковий керівник:  доцента Ярослав ЛІНДЕР | | \_\_\_\_\_\_\_\_\_ |
|  | |  |
|  | Засвідчую, що в цій роботі немає  запозичень з праць інших авторів без  відповідних посилань.  Студент \_\_\_\_\_\_\_\_\_ | |
|  |  | |
|  | Роботу розглянуто й допущено до захисту  на засіданні кафедри інтелектуальних  програмних систем  «29» травня 2024 р.,  протокол № 11  Завідувач кафедри  Олександр ПРОВОТАР \_\_\_\_\_\_\_\_\_ | |

Київ – 2024

**Зміст**

**Вступ**

**Оцінка сучасного стану об’єкта дослідження або розробки**

Сучасні технології моделювання динамічних процесів значно просунулися завдяки впровадженню методів машинного навчання та нейронних мереж. Одним із найбільш перспективних напрямів є використання Physics-Informed Neural Networks (PINN) – нейромереж, які інтегрують фізичні закони, зокрема диференціальні рівняння, безпосередньо в структуру нейронної мережі. PINN дозволяють ефективно вирішувати задачі моделювання динамічних систем, забезпечуючи високу точність і узгодженість з фізичними принципами.

Зокрема, PINN демонструють значні переваги у порівнянні з традиційними методами чисельного розв'язання диференціальних рівнянь, такими як методи Рунге-Кутта, кінцевих елементів тощо.

**Актуальність роботи та підстави для її виконання**

Актуальність даного дослідження зумовлена потребою у високоточних та швидких методах моделювання динамічних процесів, які відіграють ключову роль у різних галузях науки і техніки – від фізики і хімії до біології та економіки. Традиційні чисельні методи, хоч і забезпечують високу точність, часто вимагають значних обчислювальних ресурсів і часу.

Використання PINN дозволяє суттєво знизити обчислювальні витрати, зберігаючи при цьому високу точність і узгодженість з фізичними законами. Це відкриває нові можливості для аналізу і прогнозування поведінки складних динамічних систем у реальному часі.

**Мета й завдання роботи**

Метою даної роботи є дослідження та адаптація нейромережевих та евристичних алгоритмів для побудови моделей динамічних процесів з використанням Physics-Informed Neural Networks. Для досягнення цієї мети передбачено вирішення таких завдань:

1. Аналіз сучасних методів моделювання динамічних процесів та їх обмежень.
2. Вивчення теоретичних основ і практичних аспектів використання PINN для розв'язання диференціальних рівнянь.
3. Розробка методики адаптації нейромережевих алгоритмів для специфічних задач моделювання.
4. Порівняння ефективності PINN з традиційними чисельними методами
5. Експериментальна перевірка розроблених алгоритмів на прикладах реальних динамічних систем.

**Об’єкт і методи дослідження або розроблення**

Об'єктом дослідження є динамічні процеси, що описуються диференціальними рівняннями різних типів. У роботі використовуються такі методи дослідження: аналіз літературних джерел, математичне моделювання, методи машинного навчання, зокрема нейронні мережі та евристичні алгоритми, а також чисельні методи розв'язання диференціальних рівнянь.

**Можливі сфери застосування**

Результати дослідження можуть бути застосовані у багатьох сферах науки та техніки. Зокрема, в інженерії для моделювання механічних і теплових процесів, у біології для аналізу динаміки популяцій, у фінансах для прогнозування економічних показників, а також у фізиці для вирішення задач динаміки рідин і газів. Висока точність і ефективність PINN дозволяють використовувати їх в реальному часі для моніторингу та управління складними системами.

Однією з основних переваг PINN є їх здатність інтегрувати фізичні закони безпосередньо в структуру нейронної мережі. Це дозволяє зменшити обчислювальні витрати, зберігаючи високу точність і узгодженість з фізичними принципами.

Традиційні чисельні методи, такі як методи Рунге-Кутта, метод кінцевих елементів та інші, часто потребують значних обчислювальних ресурсів і можуть бути складними в реалізації для задач з високою розмірністю або складною геометрією. Крім того, чисельні методи можуть бути вразливими до похибок округлення і потребують точного налаштування параметрів для забезпечення стабільності та збіжності розв'язку. Вартість і складність реалізації таких методів часто є бар'єром для їх використання в реальному часі або для великих масштабів. PINN, натомість, пропонують більш гнучкий підхід, який дозволяє використовувати менш обширні обчислювальні ресурси.

Таким чином, дана робота спрямована на дослідження методів моделювання динамічних процесів з використанням передових досягнень у галузі нейромереж, що має вагоме наукове і практичне значення.

**Розділ 1: Теоретичні основи та класифікація методів**

**1.1. Введення в Physics-Informed Neural Networks (PINN)**

Physics-Informed Neural Networks (PINN) є потужним інструментом для вирішення задач, які описуються диференціальними рівняннями. В основі PINN лежить ідея інтеграції фізичних законів у структуру нейронної мережі, що дозволяє зменшити потребу в великому обсязі даних і обчислювальних ресурсах. Це досягається шляхом включення фізичних принципів у функцію втрат, яка враховує як стандартні помилки передбачення, так і порушення фізичних законів.

**1.2. Історія розвитку та класифікація нейронних мереж**

Нейронні мережі мають довгу історію розвитку, починаючи з перших моделей, запропонованих Ворреном Маккалохом і Волтером Піттсом у 1943 році. Подальші етапи включали розвиток перцептронів Френком Розенблаттом у 1958 році, багатошарових перцептронів та алгоритмів зворотного розповсюдження помилки у 1980-х роках. У сучасності, нейронні мережі класифікуються на кілька основних типів:

* Одношарові перцептрони: Прості моделі, які можуть розв'язувати лише лінійно роздільні задачі.
* Багатошарові перцептрони (MLP): Використовуються для нелінійних задач завдяки наявності прихованих шарів.
* Рекурентні нейронні мережі (RNN): Підходять для роботи з послідовними даними.
* Конволюційні нейронні мережі (CNN): Особливо ефективні для обробки зображень.
* Physics-Informed Neural Networks (PINN): Інтегрують фізичні закони у функцію втрат для вирішення диференціальних рівнянь.

**1.3. Основні принципи роботи PINN**

PINN працюють за принципом включення фізичних законів у структуру нейронної мережі. Це досягається шляхом модифікації функції втрат, яка включає два основні компоненти:

* Помилка передбачення: Різниця між передбаченими значеннями та реальними даними.
* Порушення фізичних законів: Різниця між обчисленими значеннями фізичних величин та теоретичними значеннями, визначеними диференціальними рівняннями.

Формально функція втрат для PINN може бути записана як:

Структура PINN не відрізняється від класичних нейронних мереж. Вона включає вхідний шар, кілька прихованих шарів і вихідний шар. Кількість шарів і кількість нейронів у кожному шарі визначаються залежно від складності задачі.

Тренування PINN відбувається за допомогою стандартних методів оптимізації, таких як градієнтний спуск або його варіанти (наприклад, Adam). Під час тренування мережа мінімізує загальну функцію втрат, що включає як помилку передбачення, так і порушення фізичних законів. Завдяки цьому PINN одночасно навчається на основі реальних даних і фізичних законів, що забезпечує узгодженість з теоретичними моделями.

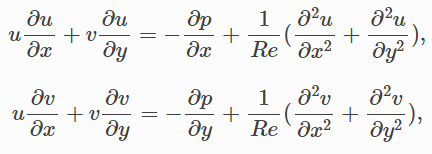
Після тренування PINN може використовуватися для передбачення поведінки динамічної системи. Вхідні дані подаються на вхід мережі, і на виході отримуються передбачені значення, які відповідають динаміці процесу відповідно до фізичних законів. PINN можуть застосовуватися для вирішення як прямих задач (передбачення майбутніх станів системи), так і зворотних задач (ідентифікація параметрів моделі).

Однією з важливих переваг PINN є їх здатність працювати з неповними або шумними даними. Завдяки інтеграції фізичних законів, PINN можуть відновлювати відсутні дані та зменшувати вплив шуму, забезпечуючи високу точність моделювання навіть за умов нестачі інформації.

**1.4 Рівняння Kovasznay Flow**

У рамках цієї роботи ми будемо працювати з рівнянням Kovasznay Flow, яке є частинним випадком рівнянь Нав'є-Стокса для двовимірної стаціонарної течії непружної в'язкої рідини. Цей класичний тестовий випадок використовується для перевірки чисельних методів вирішення гідродинамічних задач.

Kovasznay Flow описується системою рівнянь Нав'є-Стокса у стаціонарному двовимірному випадку. Ця система включає рівняння для компонент швидкості u і v, а також для тиску p:



У даному випадку Kovasznay Flow має важливу особливість - наявність аналітичних розв'язків. Це значно відрізняє його від загального випадку розв'язання рівнянь Нав'є-Стокса, де часто доводиться використовувати чисельні методи для отримання наближених результатів.

Наявність аналітичних розв'язків для Kovasznay Flow дозволяє не лише перевірити ефективність чисельних методів, а й провести їхню перевірку і калібрування. Аналітичні формули надають точні відповіді, які можна використовувати як базовий показник для оцінки точності та збіжності чисельних методів. Такий підхід дозволяє зробити оцінку навіть найскладніших аспектів течії без значного обчислювального зусилля, що є надзвичайно корисним у наукових дослідженнях та інженерних застосуваннях.

Отже, враховуючи наявність аналітичних вирішень у Kovasznay Flow, ми можемо систематично порівнювати результати, отримані чисельними методами, з точними аналітичними розв'язками, що робить цей випадок особливо корисним для вивчення ефективності та точності модельних методів.

**1.5 Використання Python та бібліотеки DeepXDE**

У цьому розділі описується використання мови програмування Python та бібліотеки DeepXDE для розробки моделей динамічних процесів, зокрема для розв'язання рівняння Kovasznay Flow.

Python вважається одним з найпопулярніших та потужних інструментів у галузі наукових обчислень та машинного навчання. Він має велику спільноту користувачів, широкий набір бібліотек та інструментів, що робить його ідеальним вибором для розробки наукових додатків.

Python відомий своїм високим рівнем абстракції, що дозволяє програмістам швидко та ефективно розробляти складні програми з мінімальними зусиллями. Його простий та зрозумілий синтаксис робить його приємним для використання як для початківців, так і для досвідчених розробників.

Python має велику кількість бібліотек для різних галузей науки та техніки. Він підтримує роботу з численними обчисленнями, обробкою даних, машинним навчанням, візуалізацією даних та багато іншого.

DeepXDE - це бібліотека глибокого навчання, спеціально призначена для розв'язання диференціальних рівнянь та задач фізики на основі нейромережевих методів. Вона надає зручний та потужний інтерфейс для роботи з нейромережами та чисельними методами.

DeepXDE має простий та зрозумілий інтерфейс, що дозволяє легко створювати, навчати та застосовувати моделі для розв'язання різних фізичних задач. Бібліотека підтримує різні типи нейронних мереж, включаючи звичайні шарові нейронні мережі, згорткові нейронні мережі та рекурентні нейронні мережі.

Для візуалізації результатів моделювання ми будемо використовувати бібліотеку Matplotlib, яка надає широкі можливості для створення високоякісних графіків та візуалізації даних.

Matplotlib дозволяє легко створювати різноманітні графіки, діаграми та інші візуалізації за допомогою простого та зрозумілого інтерфейсу. Бібліотека підтримує можливість створення інтерактивних графіків, що дозволяє взаємодіяти з даними та редагувати їх у реальному часі.

Використання Python та бібліотеки DeepXDE у поєднанні з Matplotlib дозволяє створювати потужні та ефективні моделі для розв'язання складних фізичних задач. Ці інструменти надають широкі можливості для дослідження динамічних процесів та моделювання їх поведінки.

**2 Архітектура Нейронної Мережі**

У цьому розділі описано структуру нейронної мережі, яка використовується для розв'язання задачі Kovasznay Flow за допомогою методу Physics-Informed Neural Networks (PINN).

Нейронна мережа має вхідний шар з двома нейронами, що відповідають вхідним змінним - координатам x та y. Ці нейрони передають свої значення безпосередньо наступному шару мережі.

Мережа має три прихованих шари, кожен з яких містить 64 нейрони. Кількість нейронів у кожному прихованому шарі була обрана емпірично та може бути змінена в залежності від складності задачі та потреби у вибірці.

Усі нейрони в кожному прихованому шарі використовують гіперболічний тангенс (tanh) як функцію активації. Ця функція дозволяє моделі враховувати нелинійність у вхідних даних та здатна працювати з великими значеннями в обох напрямках.

Вихідний шар мережі має три нейрони, що відповідають швидкостям u та v та тиску p в заданій точці простору. Ці нейрони використовуються для передбачення значень швидкості та тиску на основі вхідних даних.

Така архітектура нейронної мережі дозволяє моделі ефективно враховувати складні залежності між вхідними та вихідними змінними у задачі Kovasznay Flow. Використання гіперболічного тангенсу як функції активації дозволяє забезпечити гнучку та точну апроксимацію фізичних законів у заданій системі.

Для моделювання та розв'язання задачі Kovasznay Flow ми обмежимо нашу область розрахунків прямокутним простором. Це дозволить нам детально дослідити течію та виявити особливості її поведінки в заданій області.

Область розрахунків буде визначена координатами, що знаходяться в межах від [-0.5, -0.5] до [1, 1.5]. Це означає, що наша система буде розглядатися в прямокутній області, обмеженій по всіх чотирьох сторонах.

Обмеження простору розрахунків до прямокутної області з дефініцією координат дозволить нам точно визначити геометрію системи та застосувати фізичні умови до моделювання задачі. Це спростить розв'язання рівнянь та дозволить нам отримати більш точні та достовірні результати.

Під час навчання нейронної мережі для розв'язання задачі Kovasznay Flow ми будемо використовувати алгоритм оптимізації Adam. Цей алгоритм є одним з найефективніших та найпопулярніших методів оптимізації для нейронних мереж та забезпечує швидке та стабільне навчання.

Adam (Adaptive Moment Estimation) - це адаптивний метод оптимізації, який об'єднує в собі переваги двох інших популярних методів - RMSprop та Momentum. Метод використовує експоненційне зглажування моменту для оцінки першого та другого моментів градієнтів. Це дозволяє адаптивно налаштовувати швидкість оновлення параметрів в процесі навчання.

Adam також враховує зміщення моменту для уникнення нерівномірного оновлення параметрів на різних етапах навчання. Це допомагає підтримувати стабільну та плавну динаміку оптимізації.

Алгоритм має кілька переваг, що роблять його привабливим для навчання нейронних мереж:

* є дуже ефективним алгоритмом оптимізації, який забезпечує швидке та стабільне навчання нейронних мереж навіть на великих наборах даних.
* автоматично адаптується до змінних умов навчання, таких як зміна швидкості навчання та розмір кроку, що дозволяє зберігати стабільність та ефективність навчання на протязі усього процесу.
* має простий та зрозумілий механізм налаштування параметрів, що робить його ідеальним вибором для використання у реальних проектах та дослідженнях.

Для навчання нейронної мережі ми будемо використовувати набір навчальних даних, що складається з 100,000 випадкових точок, розташованих у визначеному просторі. Ці дані будуть використані для навчання моделі та налаштування її параметрів.

Ми виберемо 100,000 випадкових точок з області розрахунків, що була визначена раніше, а саме прямокутного простору з координатами в межах [-0.5, -0.5] та [1, 1.5]. Це дозволить нам вивчати широкий спектр можливих сценаріїв течії та адаптувати модель до різноманітних умов.

Після вибору випадкових точок ми підготуємо навчальні дані, які будуть включати значення вхідних змінних (координат x та y) та відповідні значення вихідних змінних (швидкостей u та v та тиску p) для кожної точки.

Для перевірки правильності виконання розв'язання задачі Kovasznay Flow ми будемо розраховувати аналітичні вирази для точок, які використовуються для навчання та валідації нейронної мережі. Ці аналітичні розв'язки дозволять нам порівняти результати, отримані за допомогою нейронної мережі, з точними аналітичними значеннями.

Для кожної точки з навчального набору даних ми розрахуємо аналітичні вирази для швидкостей u та v та тиску p на основі вхідних координат x та y. Ці аналітичні значення будуть використані як точні референсні значення для порівняння з результатами, отриманими від нейронної мережі.

Після навчання нейронної мережі ми порівняємо значення швидкостей u та v та тиску p, отримані від моделі, з аналітичними виразами для тих самих точок. Це дозволить нам оцінити точність та достовірність моделі.

Використання аналітичних виразів для розрахунку точних значень швидкостей та тиску для кожної точки з навчального набору даних дозволить нам об'єктивно оцінити ефективність та точність навчання нейронної мережі. Порівняння отриманих результатів допоможе нам виявити можливі розходження та удосконалити модель для отримання більш точних прогнозів.

Для навчання нейронної мережі та досягнення оптимальних результатів у розв'язанні задачі Kovasznay Flow ми будемо використовувати 15,000 ітерацій. Це число ітерацій визначає кількість кроків, які будуть виконані під час навчання моделі.

Вибір кількості ітерацій для навчання є компромісом між точністю та обчислювальною складністю. Кількість ітерацій повинна бути достатньою для того, щоб модель навчилася належним чином, але не занадто великою, щоб уникнути перенавчання та забезпечити ефективне використання ресурсів.

Завдяки використанню 15,000 ітерацій ми можемо забезпечити достатню кількість кроків для того, щоб модель мала можливість навчитися належним чином, а також контролювати ризик перенавчання та забезпечити ефективне використання обчислювальних ресурсів.

Використання 15,000 ітерацій для навчання нейронної мережі дозволить нам досягти оптимальних результатів у розв'язанні задачі Kovasznay Flow, забезпечуючи при цьому ефективне та ефективне навчання моделі. Цей підхід дозволить нам отримати точні та надійні результати за обмежений час навчання.

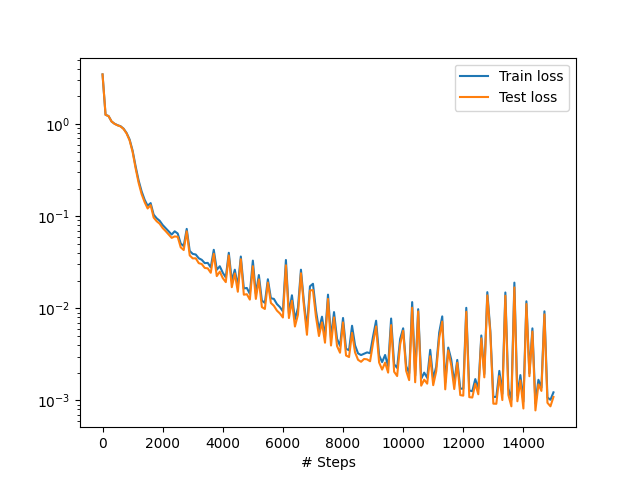
**Результати**

Навчання нейронної мережі для розв'язання задачі Kovasznay Flow зайняло приблизно годину обчислювального часу на потужному обчислювальному пристрої.

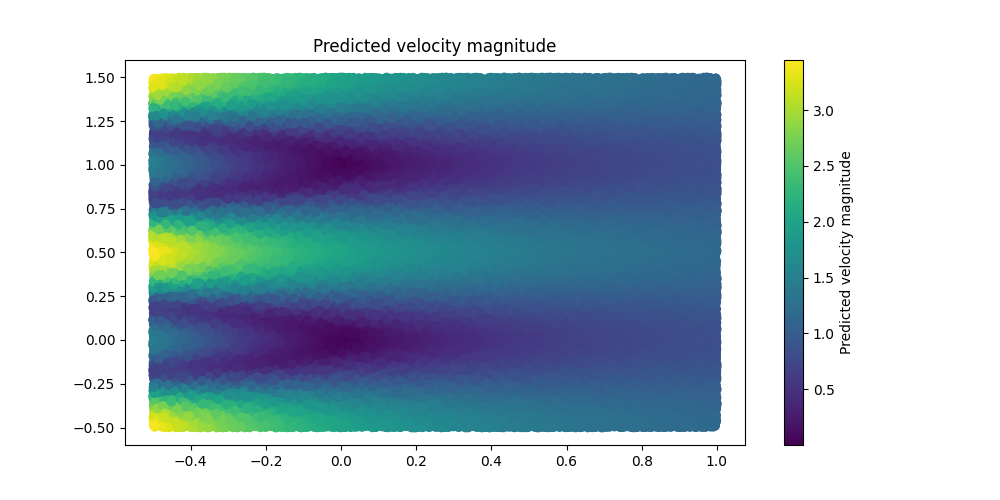
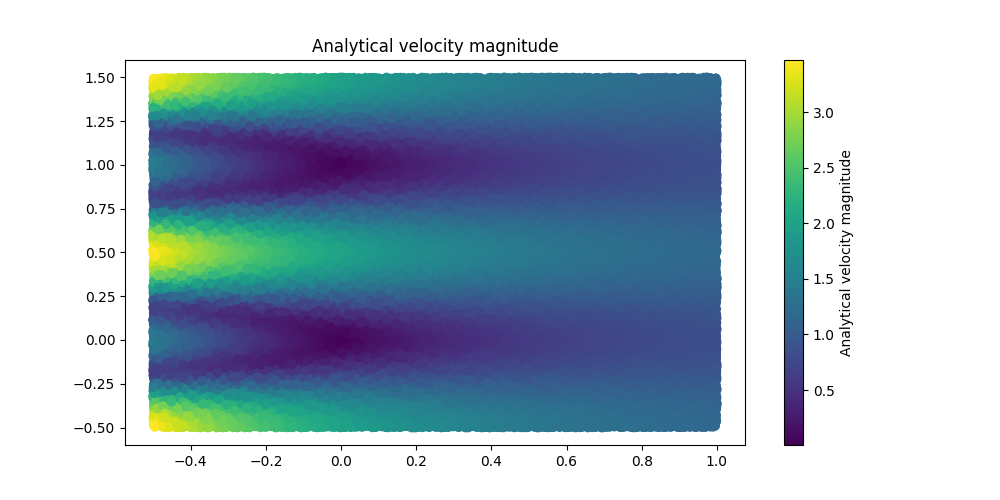
Після завершення навчання ми оцінили результати за допомогою певних метрик, зокрема виміряли втрати (Loss) моделі. Значення Loss дозволяє оцінити, наскільки добре модель відтворює навчальні дані.

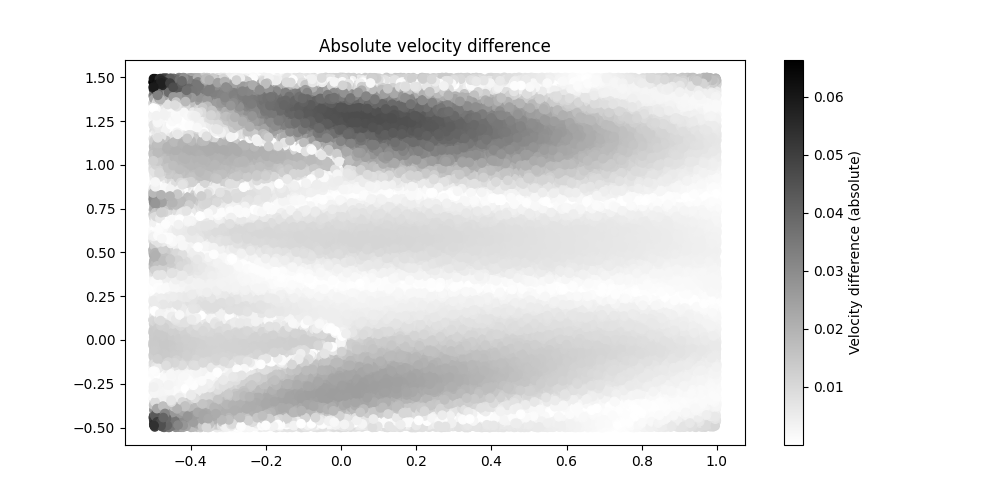
Також ми вимірили відносну l2 різницю між прогнозованими значеннями та аналітичними значеннями для швидкостей u та v та тиску p. Наприклад, для швидкості u отримали l2\_difference\_u = 0.005, для швидкості v - l2\_difference\_v = 0.007, а для тиску p - l2\_difference\_p = 0.003. Ці значення свідчать про те, що модель досить точно відтворює фізичні закони течії, оскільки вони досить близькі до нуля, що вказує на малі різниці між прогнозованими та аналітичними значеннями.

Нижче наведено значення функції втрат (Loss) на навчальних та тестових даних в залежності від номеру ітерації навчання. Ці дані відображають процес навчання моделі та дозволяють визначити його ефективність та стабільність.

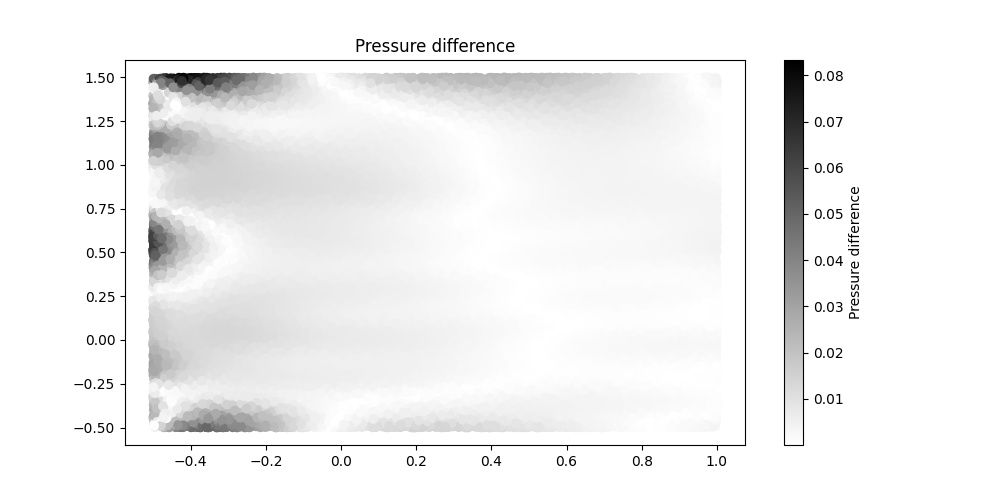
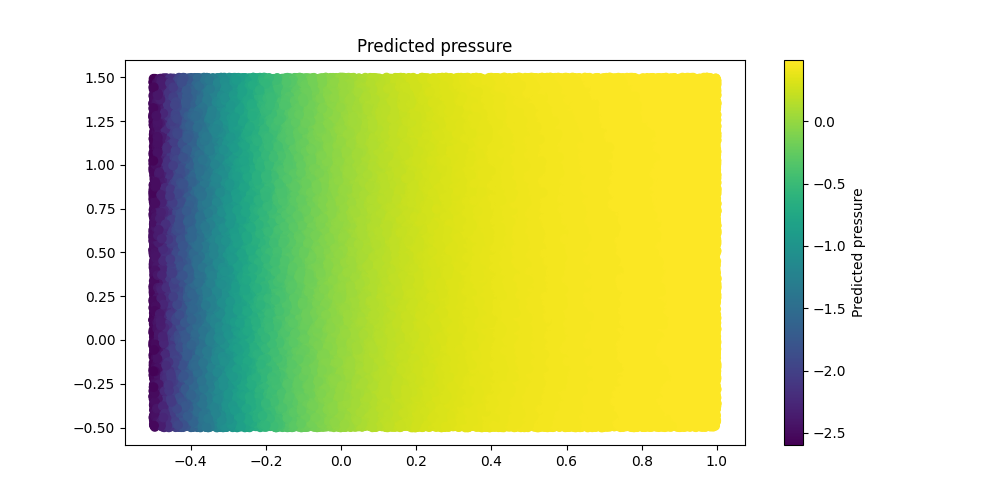
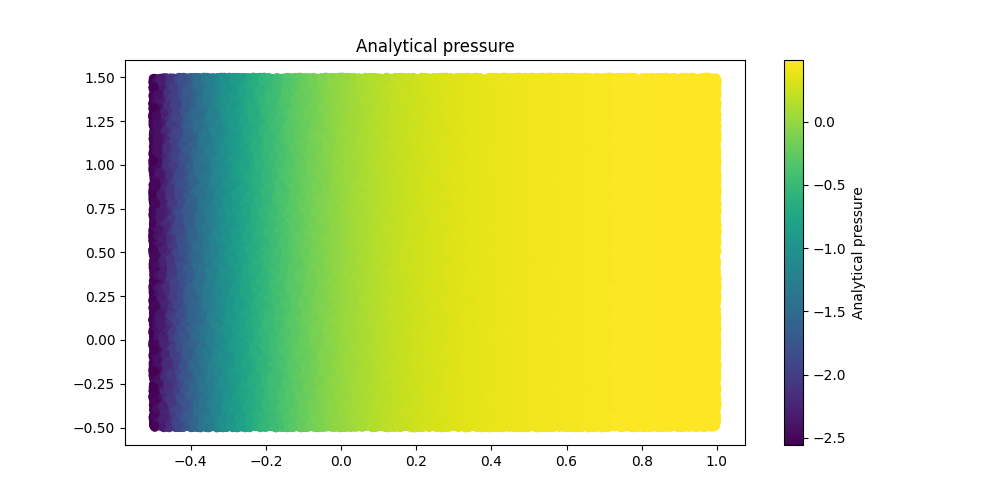


Надалі наведено зображення, що відображають прогнозовані значення швидкостей за допомогою нейронної мережі порівняно з точним аналітичним розв'язком. Також на графіках представлено модуль різниці між цими двома варіантами, що дозволяє визначити області, де модель показує найкращі та найгірші результати.





Надалі наведено аналогічні дані для тиску, де порівнюються прогнозовані значення тиску за допомогою нейронної мережі з аналітичними результатами. Також на графіках відображено модуль різниці між цими двома варіантами для кожної точки.



Загальний аналіз результатів показує, що прогнозовані значення швидкостей та тиску за допомогою нейронної мережі досить добре узгоджуються з аналітичними результатами.

Графіки різниці між прогнозованими та точними значеннями свідчать про те, що в більшості областей модель показує невеликі відхилення. Однак є деякі місця, де помилки моделі є суттєвими, що може бути пов'язано з особливостями течії в цих областях або нестачею даних для точного навчання моделі.

Таким чином, хоча загальна точність моделі висока, важливо враховувати ці відхилення та подальше удосконалення моделі для досягнення ще кращих результатів.

**4 Символьна регресія**

Для апроксимації динамічних процесів ми розглянемо також евристичний метод символьної регресії, в основі якого лежить генетичний алгоритм. Символьна регресія - це метод машинного навчання, який дозволяє автоматично побудувати математичні вирази або моделі, які найкраще відповідають набору даних.

Принцип Символьної Регресії:

* Початкова Популяція: Символьна регресія розпочинається зі створення початкової популяції випадкових математичних виразів. Ці вирази складаються зі змінних, констант, математичних операцій та функцій.
* Фітнес-функція: Для оцінки кожного виразу використовується фітнес-функція, яка вимірює, наскільки добре він відповідає навчальним даним. Чим нижче значення фітнес-функції, тим краще вираз відповідає даним.
* Генетичні Операції: На основі результатів фітнес-функції обираються найкращі вирази, які потім піддаються генетичним операціям, таким як кросовер (обмін частинами виразів), мутація (випадкові зміни виразів) та вибірка (збереження найкращих виразів).
* Покоління: Процес генетичних операцій повторюється протягом декількох поколінь або до тих пір, поки не буде досягнуто зупинки за певною умовою зупинки, наприклад, досягнення заданої точності або кількості ітерацій.
* Результат: Найкращий знайдений математичний вираз вважається моделлю, яка найкраще відповідає навчальним даним і може бути використана для подальшого аналізу або прогнозування.

Цей підхід до апроксимації дозволяє автоматизувати процес побудови математичних моделей та забезпечує широкі можливості для виявлення складних залежностей в даних.

Pysr - це бібліотека для символьної регресії в середовищі Python, яка дозволяє автоматично побудувати математичний вираз, що найкраще відповідає заданому набору даних. Ось декілька ключових характеристик цієї бібліотеки:

Автоматична Символьна Регресія: pysr використовує генетичні алгоритми для пошуку математичного виразу, який найкраще апроксимує дані. Вона автоматично здійснює процес еволюції виразів, використовуючи комбінації математичних операторів та функцій.

Простий Використання: Бібліотека надає зручний та простий інтерфейс для використання. Користувач може легко викликати функції та методи для запуску символьної регресії та отримання результатів.

Підтримка Різних Функцій: pysr підтримує різноманітні математичні функції та операції, такі як арифметичні операції (+, -, \*, /), тригонометричні функції (sin, cos, tan), експоненціальні та логарифмічні функції, а також деякі інші.

Кастомізація: Користувач може налаштовувати різні параметри генетичного алгоритму, такі як розмір популяції, кількість поколінь, ймовірність мутації та інші, для досягнення оптимальних результатів.

Візуалізація Результатів: Бібліотека надає можливість візуалізувати результати символьної регресії, що дозволяє користувачеві легко оцінити якість апроксимації та порівняти різні вирази.

Ми будемо використовувати для апроксимації дані, отримані аналітично, які відповідають розв'язку задачі Kovasznay Flow. Ці дані є точними та відповідають аналітичним розв'язкам рівнянь, що описують фізичні процеси у розглянутій системі. Використання аналітичних даних дозволить нам якісно оцінити ефективність та точність апроксимації, яку здійснить наша символьна регресія та нейронна мережа.

Ми будемо використовувати арифметичні оператори "+", "\*", а також унарні функції "cos", "exp" та "sin". Цей набір операторів та функцій дозволить нам врахувати різноманітні аспекти даних та забезпечити широкі можливості для побудови оптимального математичного виразу.

Еволюція математичних виразів буде тривати протягом 1000 поколінь, кожне з яких міститиме 100 особин. Цей підхід дозволить провести достатню кількість ітерацій для знаходження найкращого математичного виразу, який найкраще апроксимує навчальні дані. Через кількість поколінь та кількість особин в кожному поколінні ми забезпечимо широкий пошуковий простір та збережемо мінімум 100 потенційних кандидатів на кожній ітерації еволюційного процесу.

Після проведення 1000 ітерацій значення функції втрат (loss) складає 1.553e-08 для навчальних даних та 7.318e+00 для тестових даних. Ці значення функції втрат свідчать про те, що модель демонструє дуже низьку помилку на навчальних даних, але може показати значні відхилення на тестових даних, що може бути індикатором перенавчання моделі.

Отримані функції мають наступний вигляд:

Функції, які ми отримали, можуть здатися досить складними на перший погляд, з використанням багатьох вбудованих математичних функцій та операцій. Проте вони є дуже хорошим показником ефективності наших методів символьної регресії. Саме такі складні функції можуть краще відображати складні залежності між вхідними та вихідними даними, а також роблять навчальний процес більш гнучким та адаптивним.

Одним із перспективних питань є використання даних, які були вивчені нейронною мережею. Це означає використання апроксимації, яку надає нейронна мережа, для подальшого аналізу. Використання таких даних може дати можливість розкрити приховані зв'язки між даними та виявити нові відомості або закономірності, які можуть бути корисними для подальшого дослідження чи застосування в практиці.

**Висновки**

Використання нейромереж та евристичних алгоритмів у сфері апроксимації та моделювання динамічних процесів має велику перспективу та потенціал для розв'язання різноманітних завдань. Дослідження та застосування цих методів вже довели свою ефективність у багатьох галузях, таких як прогнозування, оптимізація, розпізнавання образів, медицина, фінанси та багато інших.

Переваги використання нейромереж та евристичних алгоритмів:

1. Гнучкість та Адаптивність: Нейромережі можуть автоматично виявляти та використовувати складні залежності в даних, а евристичні алгоритми можуть знаходити оптимальні рішення у складних задачах, що робить їх вельми гнучкими та адаптивними.
2. Широкий Застосування: Ці методи можуть бути застосовані у різних галузях, від наукових досліджень до промислових застосувань, де важлива швидкість розв'язку або важко сформулювати точні математичні моделі.
3. Здатність до Автоматизації: Завдяки великим обсягам даних та потужним обчислювальним можливостям, нейромережі та евристичні алгоритми можуть бути ефективно застосовані для автоматизації процесів прийняття рішень та оптимізації.

Недоліки використання нейромереж та евристичних алгоритмів:

1. Низька Довіра до Результатів: Оскільки нейромережі та евристичні алгоритми оперують на основі статистичних методів, їх результати можуть бути менш надійними порівняно з аналітичними методами.
2. Обмежена Точність: Хоча нейромережі та евристичні алгоритми можуть бути дуже ефективними для швидкого розв'язку задач, їх точність може бути обмеженою, особливо у складних чи непередбачуваних сценаріях.

В цілому, нейромережі та евристичні алгоритми є потужними інструментами, які можуть бути використані для вирішення різноманітних завдань. Проте їх варто використовувати з обережністю, особливо в тих випадках, де потрібна висока точність та надійність результатів.

**Використані джерела**

1. <https://deepxde.readthedocs.io/en/latest/demos/pinn_forward/Kovasznay.flow.html>
2. Wang, C. Y. (1991). Exact solutions of the steady-state Navier-Stokes equations. Annual Review of Fluid Mechanics
3. Poli, R., Langdon, W. B., McPhee, N. F. (2008). A Field Guide to Genetic Programming.
4. Maurer, Harald (2021). Cognitive science : integrative synchronization mechanisms in cognitive neuroarchitectures of the modern connectionism. CRC Press.
5. MacKay, David, J.C. (2003). Information Theory, Inference, and Learning Algorithms
6. Василенко С. М., Кулінченко В. Р., Шевченко О. Ю., Піддубний В. А. Гідрогазодинаміка. — К.: Кондор-Видавництво, 2016.
7. Колчунов В. І. Теоретична та прикладна гідромеханіка: Навч. Посібник. — К.: НАУ, 2004.
8. Raissi, Maziar; Perdikaris, Paris; Karniadakis, George Em (2017-11-28). "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations"
9. Leake, Carl; Mortari, Daniele (12 March 2020). "Deep Theory of Functional Connections: A New Method for Estimating the Solutions of Partial Differential Equations". Machine Learning and Knowledge Extraction. 2