ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 3 (1), 2023

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

МАТЕМАТИЧНА МОДЕЛЬ РОЗПІЗНАВАННЯ ФРАКТАЛЬНИХ СТРУКТУР З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ

О. Юнак, М. Климаш, О. Шпур, В. Мрак

Національний університет "Львівська політехніка", вул. С. Бандери, 12, Львів, 79013, Україна

Відповідальний за рукопис: Остап Юнак (e-mail: ostap.yunak@gmail.com)

(Подано 30 березня 2023)

Розглянуто методику навчання нейронної мережі розпізнавати фрактальні структури з поворотом елементів ітерації за допомогою удосконаленої рандомізованої системи ітераційних функцій. Параметри фрактальних структур є ефективним інструментом у наукових роботах, їх використовують для розрахунку складних параметрів фізичних явищ та для обчислень кількісних показників у технічних задачах. Розрахунок цих параметрів – дуже складна математична задача, оскільки дуже важко описати математичну модель фрактального зображення, визначити параметри ітераційних функцій. Навчання нейронної мережі дасть змогу швидко за готовим фрактальним зображенням визначати параметри перших ітерацій фрактала, та за їх допомогою визначати параметри ітераційних функцій. Удосконалена система рандомізованих ітераційних функцій (РСІФ) дасть змогу описати математичний процес та розробити програмне забезпечення для генерації фрактальних структур з можливостями повороту елементів ітерацій. А це дасть можливість сформувати масив даних для навчання нейронної мережі. Навчена нейронна мережа визначатиме параметри фігур перших ітерацій, на їх підставі можна буде побудувати систему ітераційних функцій, за допомогою якої можна відтворити якісно фрактальну структуру. Цей підхід застосовний для тривимірних фрактальних структур. Після встановлення параметрів перших ітерацій фрактала можна буде визначити геометричну структуру, на якій основана фрактальна структура. Цей підхід у майбутньому можна покласти в систему розпізнавання об'єктів, що містяться під фрактальними структурами, наприклад, під маскувальними сітками.

Ключові слова: *фрактал; розпізнавання фрактальних структур; рандомізована система ітераційних функцій (РСІФ); параметри ітерацій; нейронна мережа.*

1. Вступ

Параметри ітерацій фрактальної структури – це показники, які описують складність фрактала, тобто геометричної форми, яка має нерегулярну структуру та складність. Ці показники застосовують у різних галузях, серед яких:

1. Фізика: для описання складності геометричних структур, таких як гірські породи, реконструкції пульсарів, а також для вивчення фрактальної природи об'єктів у астрономії та космічних дослідженнях.

2. Біологія: для аналізу складної геометрії біологічних структур, таких як судини, рослинні листя та корені. Їх застосовують також для вивчення складних систем, таких як розгалужені системи кровоносних судин та легенів.

3. Економіка: для моделювання фінансових ринків та прогнозування їх коливань. Вони можуть допомогти в аналізі складних динамічних систем, таких як економічні системи.

4. Комп'ютерна графіка: для генерації складних геометричних фігур у комп'ютерних програмах та відеоіграх.

5. Криптографія: для захисту інформації в криптографії. Зокрема, можна використовувати фрактали для генерації випадкових ключів шифрування та створення складних шифрів.

6. Медична діагностика: для оцінювання складності структур у медичних зображеннях, наприклад, для діагностики хвороб легень, серця та мозку.

7. Матеріалознавство: для аналізу структури матеріалів, таких як метали, кераміка та полімери. Вони можуть допомогти у покращенні властивостей матеріалів та розробленні нових матеріалів зі збільшеною міцністю та еластичністю.

8. Екологія: для аналізу складної структури екосистем та вивчення екологічних процесів, таких як поширення рослинних видів у природних умовах.

9. Соціальні науки: в соціальних науках для дослідження складної динаміки соціальних систем і процесів, таких як поширення інформації, поведінкові моделі та економічні процеси.

10.Інформаційні технології: в обробленні зображень і сигналів для визначення ступеня складності структур у розподілених мереж та інше.

Узагальнюючи, фрактальну розмірність можна використовувати для аналізу та моделювання складних систем у різних галузях, вона допомагає зрозуміти їх природу та побудову.

2. Визначення рандомізованої системи ітераційних функцій для побудови фрактальних зображень

Для навчання нейронної моделі, яка б змогла розпізнавати фрактальні структури з поворотом елементів ітерації, а саме параметрів перших ітерацій, і перетворювати їх на систему ітераційних функцій РСІФ, потрібно вхідні та вихідні дані. Вхідними даними будуть фрактальні зображення (пікселі зображення), вихідними – параметри ітерацій фрактала (рис. 1).

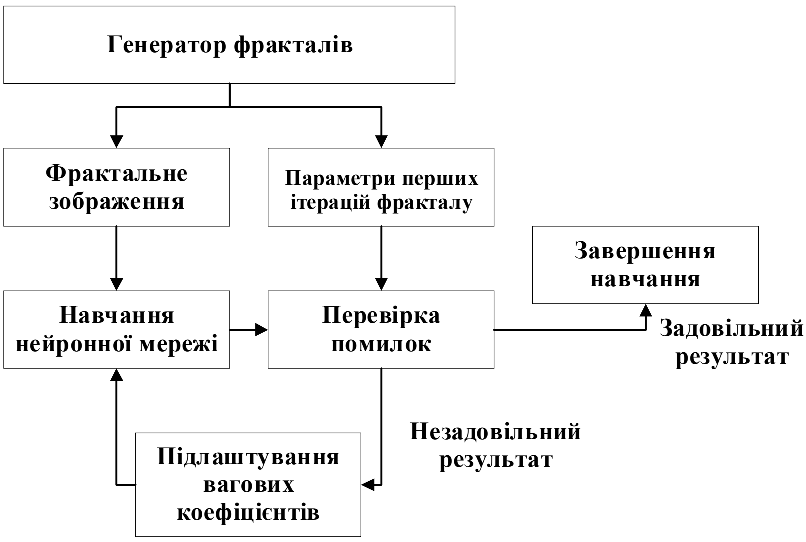

Рис. 1. Алгоритм навчання нейронної мережі

Навчання нейронної мережі потребує багато часу, а генерація фракталів є дуже складним і затратним за часом та ресурсами процесом. Генерація фрактальних зображень за допомогою РСІФ не потребує багато часу і ресурсів. Для початку опишемо відомі методи пошуку коефіцієнтів РСІФ, які використовують для побудови фрактальних зображень:

1. Експериментальний підхід: цей метод полягає в емпіричному визначенні коефіцієнтів на основі візуальної оцінки результатів побудови фрактальних зображень. Дослідник виконує ітерації з різними значеннями коефіцієнтів і оцінює отримані зображення, підбираючи коефіцієнти, які забезпечують потрібний рівень деталізації та складності зображення.

2. Метод апроксимації: використовується для визначення коефіцієнтів, які найкраще відповідають конкретному фракталу, відображаючи його побудову на основі апроксимації точок. Дослідник виконує ітерації з різними значеннями коефіцієнтів та порівнює отримані зображення із оригінальним фракталом. Коефіцієнти, що найточніше апроксимують оригінальний фрактал, вважають оптимальними.

3. Аналітичний підхід: застосовують для точного визначення коефіцієнтів, які забезпечують потрібний рівень деталізації та складності зображення. Для цього дослідник використовує математичний аналіз та теорію фракталів для розрахунку оптимальних значень коефіцієнтів.

4. Генетичний алгоритм: цей метод використовує ідеї еволюції для визначення оптимальних коефіцієнтів. Дослідник створює популяцію різних значень коефіцієнтів та виконує ітерації з кожним із них. Кожен ітераційний процес оцінюють за допомогою певної функції придатності. Найуспішніші ітерації використовують для створення нової популяції коефіцієнтів. Цей процес повторюється доти, доки не буде досягнуто оптимальних значень коефіцієнтів.

Кожний із цих методів, є складним у пошуку коефіцієнтів РСІФ, потребує складних наближених обчислень, не дає загального алгоритму пошуку коефіцієнтів для різних фракталів. Результатом виконання таких методик є СІФ (наприклад, побудови трикутника чи квадрата Серпінського) у вигляді:

$$\left[\begin{array}{l} \begin{cases} x_n = a_1 + b_1 x_{n-1} + c_1 y_{n-1} \\ y_n = d_1 + e_1 y_{n-1} + f_1 y_{n-1} \end{cases} \\ \ldots \\ \begin{cases} x_n = a_j + b_j x_{n-1} + c_j y_{n-1} \\ y_n = d_j + e_j y_{n-1} + f_j y_{n-1} \end{cases} \end{array}\right. , \tag{1}$$

де $\{a_1, b_1, c_1, d_1, e_1, f_1 .... a_j, b_j, c_j, d_j, e_j, f_j\}$, $j \in N$ – коефіцієнти ітераційних функцій, які визначають вигляд фігур першої ітерації;

Така система ітераційних функцій малоінформативна, оскільки система не дає змоги визначити центри фігур першої ітерації фрактала, ширину і довжину, коефіцієнти пропорційностей та кути повороту, що, своєю чергою, не дає можливості визначити параметри перших ітерацій фрактала.

Для вирішення цієї проблеми візьмемо за основу удосконалену РСІФ [4]:

$$\left[\begin{array}{l} \begin{cases} x_n = x_{a_1} - \dfrac{x_{a_1} - x_{n-1}}{k_{Xa}} \\ y_n = y_{a_1} - \dfrac{y_{a_1} - y_{n-1}}{k_{Ya_1}} \end{cases} \\ \qquad \ldots \\ \begin{cases} x_n = x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{X_{a_j}}} \\ y_n = y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Y_{a_j}}} \end{cases} \end{array}\right. , \tag{2}$$

де $(x_n \ , y_n)$ – поточна координати точки, яка покроково будує фрактал; $\{(x_{a_1}, \ y_{a_1}) \dots (x_{a_j}, \ y_{a_j})\}$ – координати точок, що утворюють перші ітерації фрактала [4]; $\{k_{Xa_1}, \ k_{Ya_1} \dots \ k_{Xa_j}, \ k_{Ya_j}\}$ – коефіцієнти пропорційності відповідних фігур першої ітерації [4].

Для визначення коефіцієнтівпропорційності відповідних фігур першої ітерації k_{Xa}, $k_{Ya} \dots$ k_{Xb}, k_{Yb} використаємо таку формулу [4]:

$$k = \frac{L}{l}, \tag{3}$$

де k – коефіцієнт пропорційності відповідної фігури першої ітерації фрактала; L – початкова довжина або ширина фрактального зображення; l – початкова довжина або ширина зображення відповідної фігури першої ітерації фрактала.

Для визначення координат точок, що утворюють перші ітерації фрактала (x_{a_j}, y_{a_j}), використаємо центри фігур першої ітерації X_{a_j}, Y_{a_j} [4]:

$$x_{a_j} = \frac{k_{Xa_j} X_{a_j} - \frac{L_x}{2}}{k_{Xa_j} - 1}; \ y_{a_j} = \frac{k_{Ya_j} Y_{a_j} - \frac{L_y}{2}}{k_{Ya_j} - 1}. \tag{4}$$

Запропонована система ітераційних функцій РСІФ (1) не дає можливості описати фрактал з поворотними елементами перших ітерацій. Для розв’язання цієї задачі візьмемо одну пару ітераційних функцій (5):

$$\begin{cases} x_n = x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} \\ y_n = y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} \end{cases}. \tag{5}$$

Паралельним перенесення перенесемо центр фігури першої ітерації у початок центру координат та подамо систему пари ітераційних функцій так (6):

$$\begin{cases} x_n = x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \\ y_n = y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \end{cases}, \tag{6}$$

де X_{a_j}, Y_{a_j} – центри фігур першої ітерації;

Тепер застосуємо афінні перетворення повороту на кут $\boldsymbol{\varphi}$ для формули (6):

$$\begin{cases} x_n = \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \cos(\phi_j) - \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \sin(\phi_j) \\ y_n = \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \sin(\phi_j) + \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \cos(\phi_j) \end{cases}, \tag{7}$$

Наступним кроком паралельним перенесенням перемістимо центр фігури першої ітерації туди, де він був розмішений перед тим (8):

$$\begin{cases} x_n =_n = \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \cos(\phi_j) - \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \sin(\phi_j) + X_{a_j} \\ y_n = \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \sin(\phi_j) + \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \cos(\phi_j) + Y_{a_j} \end{cases}. \tag{8}$$

Використовуючи формулу (8), подамо загальну систему ітераційних функцій у такому вигляді (9):

$$\left[\begin{array}{l} \left\{\begin{array}{l} x_n = \left(x_{a_1} - \dfrac{x_{a_1} - x_{n-1}}{k_{Xa_1}} - X_{a_1} \right) \cdot \cos(\phi_1) - \left(y_{a_1} - \dfrac{y_{a_1} - y_{n-1}}{k_{Ya_1}} - Y_{a1} \right) \cdot \sin(\phi_1) + X_{a_1} \\ y_n = \left(x_{a_1} - \dfrac{x_{a_1} - x_{n-1}}{k_{Xa_1}} - X_{a_1} \right) \cdot \sin(\phi_1) + \left(y_{a_1} - \dfrac{y_{a_1} - y_{n-1}}{k_{Ya_1}} - Y_{a_1} \right) \cdot \cos(\phi_1) + Y_{a_1} \end{array}\right. \\ \qquad \ldots \\ \left\{\begin{array}{l} x \;= \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \cos(\phi_j) - \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \sin(\phi_j) + X_{a_j} \\ y_n = \left(x_{a_j} - \dfrac{x_{a_j} - x_{n-1}}{k_{Xa_j}} - X_{a_j} \right) \cdot \sin(\phi_j) + \left(y_{a_j} - \dfrac{y_{a_j} - y_{n-1}}{k_{Ya_j}} - Y_{a_j} \right) \cdot \cos(\phi_j) + Y_{a_j} \end{array}\right. \end{array}\right. , \qquad (9)$$

Побудуємо фрактал трикутник Серпінського за допомогою удосконаленої РСІФ (8) та виведемо результат на рис. 2.

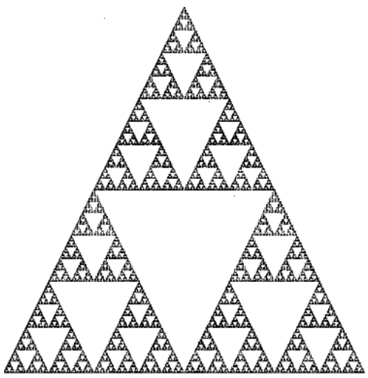

Рис. 2. Трикутник Серпінського, побудований за допомогою РСІФ (8)

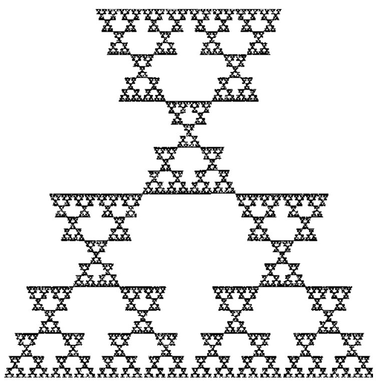

Рис. 3. Трикутник Серпінського, побудований за допомогою РСІФ (9) з поворотом фігури першої ітерації на 180^0

Змінимо кут ϕ для фігури першої ітерації на 180^0 та за допомогою удосконаленої РСІФ (9) побудуємо фрактал і виведемо результат на рис. 3.

3. Алгоритм роботи генератора фракталів на базі удосконаленої РСІФ

Алгоритм генератора фрактальних зображень на базі РСІФ подано на рис. 4. Його особливість – урахування кута повороту фігури першої ітерації ϕ_j та визначення координат фрактальних зображень з використанням удосконаленої системи ітераційних функцій (9). Запропонований алгоритм дає змогу генерувати фрактали з поворотом елементів першої ітерації, що не враховано у формулі (2).

Щоб навчити нейронну мережу визначати розмірність фрактального зображення, необхідно згенерувати фрактал із заданими параметрами фігур першої ітерації. Вхідними даними будуть фрактальні зображення (пікселі зображення), вихідними даними – параметри фігур першої ітерації, за якими будемо будувати систему удосконалену систему РСІФ (9) (рис. 1).

Для програмної реалізації нейронної мережі використаємо мову програмування JavaScript та технології WEB-програмування для візуалізації. У JavaScript є готові бібліотеки для навчання нейронної мережі – brain.js

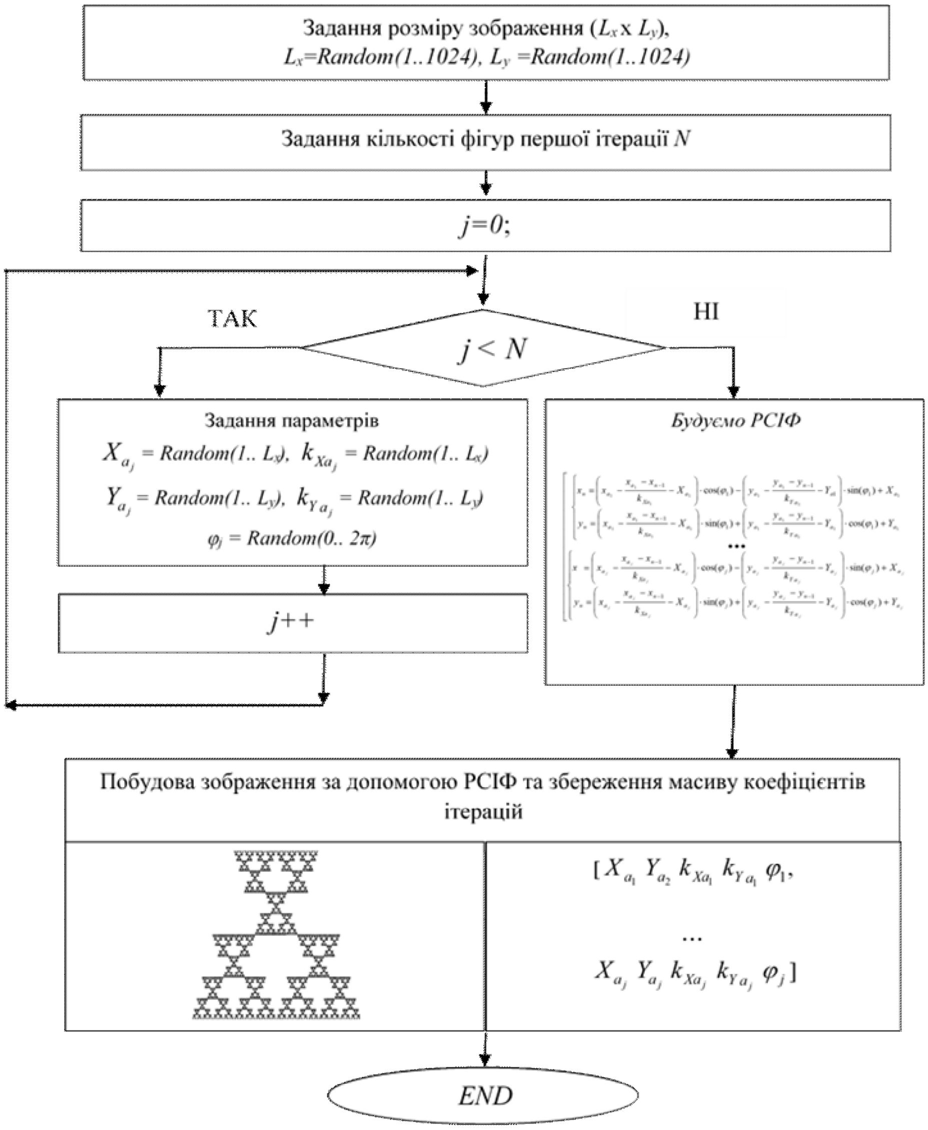

Рис. 4. Алгоритм роботи генератора фракталів на базі удосконаленої РСІФ

Бібліотека brain.js – це бібліотека для навчання нейронних мереж на JavaScript. Вона дає змогу створювати різноманітні типи нейронних мереж, зокрема персептрон, LSTM та RNN. Brain.js забезпечує зручний інтерфейс для створення, навчання та використання нейронних мереж. Вона дає змогу зберігати навчені моделі в форматі JSON, що дозволяє легко зберігати та повторно використовувати їх. Однією із найважливіших особливостей brain.js є її можливості працювати в браузері та на сервері, що дає змогу використовувати навчені моделі у різних середовищах. Brain.js – потужна та гнучка бібліотека для створення та навчання нейронних мереж на JavaScript, що дозволяє легко і швидко створювати та використовувати навчені моделі.

Для вирішення нашого завдання використовуватимемо тип мережі персептрон, а саме багатошаровий перцептрон Румельхарта, оскільки основу вхідних і вихідних даних становитиме фрактальне зображення (що, своєю чергою, вже потребує багато прихованих шарів) та масив коефіцієнтів першої ітерації фрактала. Для навчання персептронів у бібліотеці brain.js використаємо алгоритм зворотного поширення помилки (*backpropagation*). Він дає змогу коригувати ваги нейронів у зворотному напрямку від виходу до входу мережі, зменшуючи помилку між прогнозованим та правильними значеннями.

Конфігурацію нейронної мережі на основі бібліотеки brain.js наведено в табл. 1.

Таблиця 1

Конфігурація нейронної мережі на основі бібліотеки brain.js для навчання за допомогою системи РСІФ

Параметр	Значення
Тип нейронної мережі	багатошаровий перцептрон Румельхарта
Алгоритм навчання мережі	алгоритм зворотного поширення помилки (backrpropagation)
Кількість шарів персептронів	з довільною кількістю шарів та нейронів у кожному з них (автопідлаштування)

Обмеження, які виникає під час роботи бібліотеки brain.js – необхідність мати достатньо велику кількість даних для навчання нейронної мережі, щоб вона могла ефективно працювати. Якщо в даних занадто багато шуму або даних недостатньо, то може виникнути проблема перенавчання (*overfitting*). Інші обмеження, які виникають під час роботи бібліотеки brain.js – кількість ітерацій (*epochs*) визначає, скільки разів модель повторює процес навчання на навчальних даних. Занадто мала кількість ітерацій може призвести до недостатнього навчання, а занадто велика – до перенавчання. Це завдання вирішує генератор РСІФ (рис. 4): дані, які надходитимуть у вигляді зображень, не міститимуть шуму, що унеможливлює проблему перенавчання (*overfitting*).

Схему програмної реалізації зображено на рис. 5.

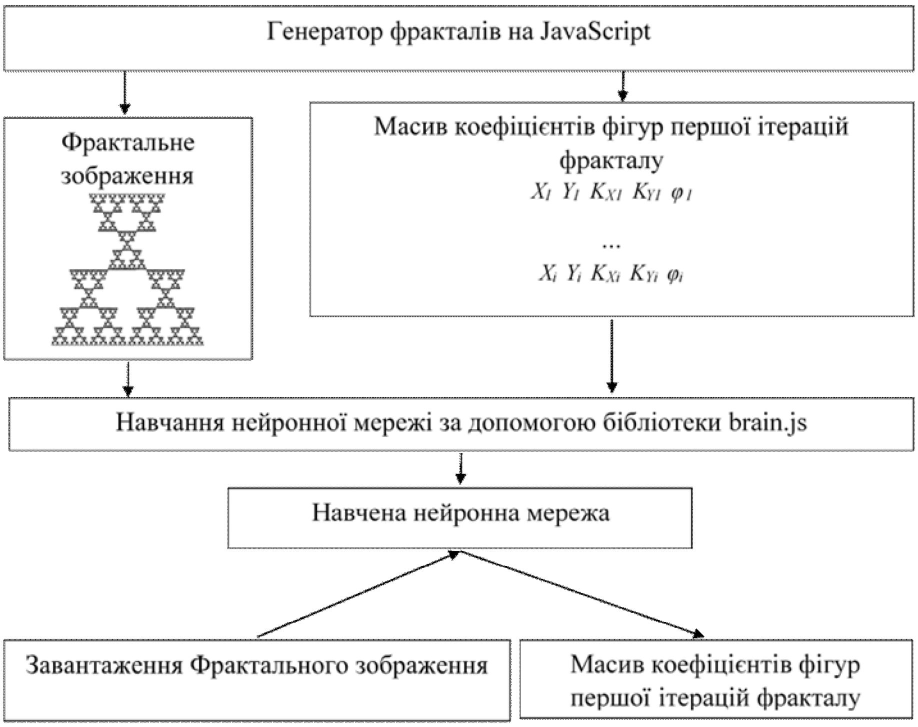

Рис. 5. Програмна реалізація нейронної мережі на базі бібліотеки Brain.js

4. Дослідження ефективності програмної реалізації алгоритму

Після написання коду програми алгоритм генерації фракталів на базі удосконаленої РСІФ і навчання нейронної мережі, перевіримо результати роботи.

Після завантаження у програму трикутника Серпінського, всі фігури першої ітерації якого повернуті на кут 180^0 (рис. 6), нейрона мережа видала результат з параметрами фігур першої ітерації фрактала, наведеними у табл. 2.

Для наочнішого оцінювання роботи алгоритму виконаємо математичні розрахунки та знайдемо параметри перших ітерацій фрактала із рис. 6, і порівняємо їх з тими, що дала навчена нейронна мережа (табл. 3).

Таблиця 2

Результати роботи навченої нейронної мережі

Номер ітераційної фігури	L_x	L_y	X_i	Y_i	K_{Xi}	K_{Yi}	φ_i
1	500,0012	499,9962	250,0011	125,0007	1,9987	1,9973	180,0009
2			125,0013	375,0011	2,001	2,0012	179,9991
3			374,9989	375,0009	2,0012	1,9985	179,999

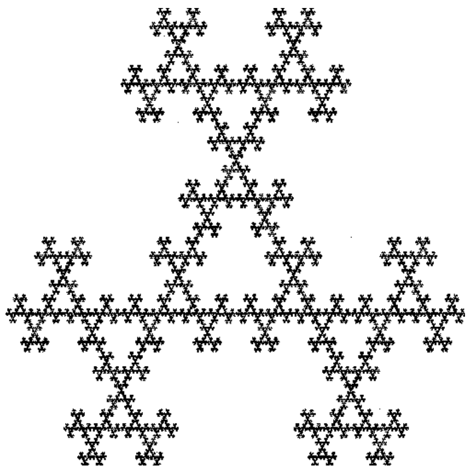

Рис. 6. Трикутник Серпінського
з поворотом усіх фігур першої ітерації на 180^0

Таблиця 3

Результати розрахунків параметрів перших ітерацій фрактала (рис. 6)

Номер ітераційної фігури	L_x	L_y	X_i	Y_i	K_{Xi}	K_{Yi}	φ_i
1	500	500	250	125	2	2	180
2			125	375	2	2	180
3			375	375	2	2	180

Похибка обчислення нейронною мережею – менше ніж 0,1 %, що свідчить про доволі високу точність визначення параметрів перших ітерацій фрактала та застосування удосконаленої РСІФ для навчання нейронної мережі.

Висновки

Визначення параметрів ітерації фрактальної структури – складна математична задача, навчена нейронна мережа дає змогу виконувати її. Запропонований алгоритм генерації фракталів за допомогою удосконаленої рандомізованої системи ітераційних функцій дає можливість автоматизувати навчання нейронної мережі. Навчена нейронна мережа видає параметри ітерацій фрактальної структури з дуже високою точністю, похибка обчислення нейронною мережею менша за 0,1 %. Це рішення дає змогу за допомогою визначених параметрів побудувати удосконалену рандомізовану систему ітераційних функцій, розробити програмний конструкт формування фрактального зображення із поворотами ітерацій. Маючи параметри перших ітерацій фрактала, можна визначити геометричну структуру, яка є основою фрактальної структури. Цей підхід у майбутньому може бути покладений в основу системи розпізнавання об’єктів, що містяться під фрактальними структурами, наприклад, розпізнавання об’єкта під маскувальними сітками, що важливо для військової сфери.

Список використаних джерел

[1] *Al-shameri, W. F. H. Deterministic algorithm for constructing fractal attractors of iterated function systems. Eur. J. Sci. Res. 2015, 134, pp. 121–131.*

[2] *Mandelbrot, B. B. The Fractal Geometry of Nature; W. H. Freeman & Company: New York, NY, USA, 1999.*

[3] *O. Yunak, O. Shpur, B. Strykhaliuk, M. Klymash. Algorithm forming randomized system of iterative functions by based cantor structure. Information and communication technologies, electronic engineering, 2021, No. 1 (2), pp. 71–80.*

[4] *M. C. Gutzwiller, Benoît B. Mandelbrot, C. J. G. Evertsz, et al. Fractals and Chaos: The Mandelbrot Set and Beyond. Springer New York, 2010. ISBN: 1441918973.*

[5] *B. B. Mandelbrot. The Fractal Geometry of Nature; Echo Point Books & Media, LLC, 2021. 490 p. ISBN-10: 1648370403.*

[6] *Z. Z. Falconer, Kenneth Falconer. Techniques in Fractal Geometry. Wiley & Sons, Incorporated, John. 1997. 274 p. ISBN: 0471957240.*

[7] *Юнак О. М., Пелещак Б. М., Охремчук Н. Л., Метлевич Я. Р. Перетворення зображення фрактальної структури типу "Фрактальний пил" (множина Кантора) в рандомізовану систему ітераційних фунцій, XII Міжнар. наук.-практ. конференція "Последните постижения на Европейската наука - 2016", Том 13, София "Бял ГРАД-БГ" ООД, 2016. 90 с.*

[8] *Mandelbrot, B. B. Fractals: Form, Chance and Dimension, Echo Point Books & Media; Reprint ed. edition 2020. 656 p.*

[9] *Kenneth Falconer, Fractal Geometry: Mathematical Foundations and Applications 3rd Edition, 2014. 400 с. ISBN-10: 111994239X.*

[10] *The Mandelbrot Set and Beyond New York: Springer, 2004. 308 p. ISBN: 0-387-20158-0.*

[11] *Peter R. Massopust. Fractal Functions, Fractal Surfaces, and Wavelets. Elsevier Science & Technology. Elsevier Science & Technology, 1995. 383 p. ISBN: 0124788408.*

MATHEMATICAL MODEL OF FRACTAL STRUCTURES RECOGNITION USING NEURAL NETWORK TECHNOLOGY

Ostap Yunak, Mykhailo Klymash, Olha Shpur, Vasyl Mrak

Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013, Ukraine

The article goes about the methods of training a neural network to recognize fractal structures with the rotation of iteration elements by means of an improved randomized system of iteration functions. Parameters of fractal structures are used to calculate complex parameters of physical phenomena. They are an effective tool in scientific works and used to calculate quantitative indicators in technical tasks. The calculation of these parameters is a very difficult mathematical problem. This is caused by the fact that it is very difficult to describe the mathematical model of the fractal image, it is difficult to determine the parameters of the iterative functions. The neural network learning will allow you to quickly determine the parameters of the first iterations of the fractal based on the finished fractal image and basing on them to determine the parameters of the iterative functions. The improved system of randomized iterative functions (SRIF) will allow to describe the mathematical process and to develop the software for generating fractal structures with the possibility of rotating elements of iterations. In its turn, this will make it possible to form an array of data for training a neural network. The trained neural network will be able to determine the parameters of the figures of the first iterations by means of which it will be possible to build a system of iterative functions. It will help to reproduce a fractal structure qualitatively. This approach can be used for three-dimensional fractal structures. After setting the parameters of the first iterations of the fractal, it will be possible to determine the geometric structure which is the basis of the fractal structure. In the future, this approach may be included in the system for recognizing objects under fractal structures, for example, under masking nets.

Key words: *fractal; recognition of fractal structures; randomized system of iterative functions (RSIF); parameters of iterations; neural network.*