

ОПТИМІЗАЦІЯ АЛГОРИТМУ РОБОТИ НЕЙРОННОЇ МЕРЕЖІ ЗА РАХУНОК ЗМЕНШЕННЯ ШИРИНИ ПОТОКОВОГО ГРАФА АЛГОРИТМУ

Фастюк Євген Ігорович

ORCID ID: 0009-0005-9225-5725

аспірант кафедри електронних обчислювальних машин
Національний університет «Львівська політехніка», Україна

Гузинець Наталія Вікторівна

асистент кафедри електронних обчислювальних машин
Національний університет «Львівська політехніка», Україна

В сучасному світі, коли обчислювальна потужність є критичним ресурсом, оптимізація алгоритмів стає важливим завданням для забезпечення ефективної роботи програмних систем. Одним із ключових напрямів у цьому контексті є зменшення витрат ресурсів, зокрема кількості процесорних елементів (ПЕ), необхідних для виконання алгоритмів. У цьому контексті важливо дослідження зменшення ширини потокового графа алгоритму нейронної мережі. Нейронні мережі стали потужним інструментом у багатьох областях, від розпізнавання образів до автономних систем. Проте їх ефективність і потужність може бути обмежена необхідністю великої кількості обчислювальних ресурсів. Тому питання оптимізації роботи нейронних мереж стає вкрай актуальним.

Як приклад візьмемо нейронну мережу Deep Feed-Forward (DFF) – глибинну мережу прямого зв'язку. Таку мережу можна представити потоковим графом, де вузлами будуть шари нейронів, а ребрами - зв'язки між ними. На початку графу буде вузол вхідного шару, де кожен вхідний приклад буде входити до мережі. Потім вихід з кожного нейрона вхідного шару буде передаватися через ребра до першого прихованого шару, де виконуватимуться різні операції, зазвичай з використанням функції активації. Після обробки даних в кожному прихованому шарі, вихід буде передано до наступного прихованого шару, де процес повториться [1]. Граф глибинної мережі прямого зв'язку (рис. 1) виглядатиме так:

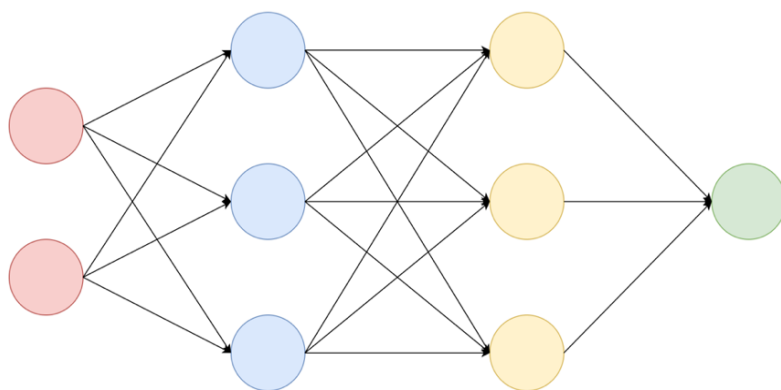


Рис. 1. Граф глибинної мережі прямого зв'язку

Для зображення та дослідження графа алгоритму G необхідно точно описати множини вершин K та дуг R графа алгоритму (рис. 2) [2].

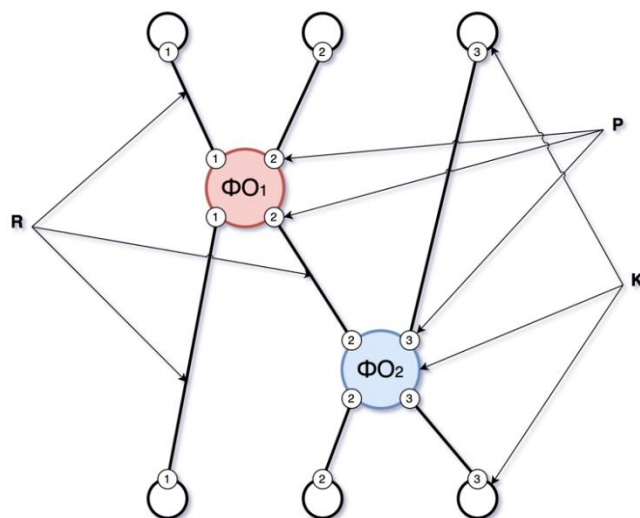


Рис. 2. Граф алгоритму $G(K, P, R)$

Розбиваємо граф нейронної мережі на менші функціональні оператори (ФО). При побудові потокового графу алгоритму було використано базовий ФО з двома входами та двома виходами. Після розбиття будуємо потоковий граф алгоритму (ПГА) глибинної нейронної мережі прямого зв'язку та представляємо потоковий граф алгоритму у формі структурної матриці. Структурна матриця F розмірністю $l \times n$, елементами якої є номери ФО, присвоєні кожній дузі, по якій на ФО надходять операнди. Нумерація починається від 1 і призначається всім наступним ФО підряд в порядку зростання. Решта елементів такої матриці заповнюються нулями.

За допомогою структурної матриці, визначаємо характеристики та показники сформованого потокового графа алгоритму [3]:

- Кількість аргументів алгоритму;
- Кількість використовуваних у i -й момент часу процесорних елементів w_i для реалізації паралельного алгоритму;
- Час виконання послідовного алгоритму T_l ;
- Час виконання паралельного алгоритму T_p ;
- Кількість необхідних процесорних елементів p для максимального розпаралелювання алгоритму;
- Прискорення S_p ;
- Ефективність E_p .

На основі отриманих даних проводимо аналіз завантаженості процесорних елементів, зменшення ширини потокового графа алгоритму, та перебудову потокового графа алгоритму з врахуванням можливої зміни необхідної кількості процесорних елементів. Коефіцієнт завантаженості ПЕ визначається відношенням загальної кількості ПЕ, до кількості ПЕ необхідних для виконання нейронної мережі.

Після перебудови потокового графа алгоритму знову визначаємо характеристики та показники сформованого потокового графа алгоритму, визначаємо коефіцієнт завантаженості процесорних елементів та порівнюємо показники до та після зменшення ширини потокового графа алгоритму.

Висновки. За допомогою отриманих даних вдалося зменшити ширину потокового графа алгоритму нейронної мережі з 3 до 2 зі збереженням його

ефективності. Отриманий результат може бути використаний для побудови нейронної мережі та її виконання на одноплатних мікрокомп'ютерах, таких як Raspberry Pi, оскільки такі комп'ютери стали ключовим фактором у популяризації вбудованих систем та розвитку IoT (інтернет речей) та є надзвичайно важливими для забезпечення ефективної роботи сучасних програмних систем, де такі мікрокомп'ютери застосовуються [4].

Список використаних джерел:

1. Ayyadevara V. K. *Neural Networks with Keras Cookbook: Over 70 Recipes Leveraging Deep Learning Techniques Across Image, Text, Audio, and Game Bots*. Packt Publishing, Limited, 2019.
2. Мельник А. О., Яковлева І. Д., Ющенко В. Ю. ПОБУДОВА ТА МАТРИЧНЕ ПОДАННЯ ПОТОКОВОГО ГРАФА АЛГОРИТМУ. Вісник Вінницького політехнічного інституту. 2009. № 3. С. 93–99.
3. Мельник А. О., Яковлева І. Д. ПОДАННЯ ПОТОКОВОГО ГРАФА АЛГОРИТМУ СТРУКТУРНОЮ МАТРИЦЕЮ. Вісник Хмельницького національного університету. 2008. № 4. С. 124–128.
4. Fastiuk Y., Bachynskyi R., Huzynets N. Methods of Vehicle Recognition and Detecting Traffic Rules Violations on Motion Picture Based on OpenCV Framework. *Advances in Cyber-Physical Systems*. 2021. Vol. 6, no. 2. P. 105–111. URL: <https://doi.org/10.23939/acps2021.02.105> (date of access: 26.05.2024).