УД 004.932.1; 004.932.2; 004.932.4.

П.О. Приставка, ДТН, А.К. Шевченко
(Національний Авіаційний Університет, Україна)

Дослідження можливого підвищення швидкодії операції згортки за допомогою AArch64 SIMD.

Запропоновано метод оптимізації операції двовимірної згортки (СО) за допомогою 16-розрядних SIMD технологій для ARMx64 (aarch64). Проведено порівняльний аналіз із СО, що реалізовано у бібліотеці OpenCV.

Вступ. Обробка цифрових зображень (ЦЗ) грає важливу роль в розробці програмного забезпечення (ПЗ), такого як: обробка відеопотоку (стабілізація, фільтрація, зменшення шуму, стиснення тощо), обробка одного зображення у задачах машинного навчання. Велика кількість задач обробки ЦЗ може бути зведена до певної форми операції згортки (ОЗ) або передбачає її використання:

$$p_{i,j} = \sum_{k=0}^{r} \sum_{l=0}^{c} \Gamma_{k,l} P_{k+i-a,l+j-a'}, \qquad (1)$$

де $i = a,\ldots,W-(r-a)-1$, $j = a',\ldots,H-(c-a')-1$ індекси пікселів ЦЗ p; W та H є розмірами (шириною та вишиною) P джерела та p результуючого зображення, Γ є матрицею згортки (МЗ) (розміром $r \times c$), a та a' є "якорями" що визначають взаємне положення відфільтрованої точки щодо МЗ.

Короткий огляд сучасної оптимізації ПЗ. Одним із підходів до підвищення продуктивності обробки ЦЗ є використання різних бібліотек [1][2], таких як OpenCV та ARM Compute Library (ACL). Вони містять оптимізований код не тільки для ARM – NEON (для архітектур ЦП armeaby-v7a та arm64-v8a). Ще одна розумна стратегія полягає у використанні колекції бібліотек, об'єднаних в єдину структуру, щоб переваги однієї бібліотеки компенсували недоліки інших. OpenCV [3] і ACL [4] є гарними прикладами бібліотек, що містять широкий спектр алгоритмів, включаючи обробку DI та аналіз DI. Крім того вони містить модулі для навчання CNN, оптимізовані за допомогою SIMD (різних архітектур ЦП). Крім того, OpenCV добре відомий своєю високоякісною обробкою ЦЗ. Тому ми розглядатимемо OpenCV як еталон для порівняння

Основною перешкодою для використання оптимізації SIMD є те, що операції SIMD виконуються над цілими числами малого об'єму (не більше 16-bit), тому ми повинні перетворити записи з плаваючою комою МЗ у цілочисельний вигляд.

Представимо елементи ядра з (1) у відповідному вигляді:

$$\Gamma_{i,j} = \nu\gamma_{i,j}, \quad \nu \in \mathbb{R}, \quad \gamma_{i,j} \in \mathbb{Z} \qquad (2)$$

де ν – коефіцієнт нормалізації. Зауважимо, що рівняння (1) є досить загальним і цілком сумісним із функцією cv::filter2D(...) бібліотеки OpenCV [3]. Зауважимо

що надалі ми будемо використовувати квадратні МЗ, тобто $r = c$. Отже після найбільш ресурсномісткої операції можна виконати нормалізацію результату.

Будь-яке ядро можна представити у формі (2), але чим точніший результат ми хочемо, тим більшій розрядності має бути $\gamma_{i,j}$. Таким чином, ми повинні встановити деякі обмеження, яких слід виконувати, щоб не виходити за межі 16-bit діапазону числа коли буде виконуватися ОЗ:

$$0 \le (2^8 - 1) \times \sum_{i=0}^{r} \sum_{j=0}^{r} \gamma_{i,j} \le 2^{16} - 1. \quad (3)$$

Якщо ядро містить негативні елементи, умова має бути дещо складнішою:

$$0 \le (2^8 - 1) \times \sum_{i=0}^{r} \sum_{j=0}^{r} |\gamma_{i,j}| \le 2^{16-1} - 1, \quad (4)$$

Ми пропонуємо вибрати для заданого Γ найбільший можливе ν, щоб (3) або (4) все ще задовольнялися. І якщо так, то можна побачити, що безліч корисних ядер можна привести до відповідної форми. У поточній доповіді ми пропонуємо новий метод оптимізації операції згортання (СО), що застосує SIMD NEON64.

Опис та результати експерименту. Як еталон обрали функції cv::filter2d(...) бібліотеки OpenCV. Останній добре відомий серед дослідників AI та ОЦЗ завдяки високоякісному та оптимізованому коду. Для порівняння ми використали останній доступний стабільний тег бібліотеки OpenCV (на момент дослідження) - 4.6.0 (2022-06-05). Процедуру проводили для різних розмірів ядер: 2×2, 3×3, …, 15×15.

Таблиця 1.

Типові значення та варіабельність виконання ОЗ cv::filetr2D(…)

M	$\bar{x}$	*MED*	*S*	*W*
2	0,09490	0,09478	0,00019	0,00197
3	0,17566	0,17575	0,00024	0,00139
4	0,28537	0,28521	0,00013	0,00047
5	0,43033	0,43033	0,00020	0,00047
6	0,60653	0,60632	0,00071	0,00117
7	0,81523	0,81541	0,00024	0,00030
8	1,81854	1,82405	0,01654	0,00910
9	1,77349	1,77476	0,00309	0,00174
10	1,87042	1,86961	0,00698	0,00373
11	1,87069	1,87164	0,00335	0,00179
12	1,87757	1,87769	0,00274	0,00146
13	1,88953	1,88958	0,00249	0,00132
14	1,88621	1,88689	0,00397	0,00210
15	1,89801	1,89551	0,00880	0,00464

Таблиця 2.

Типові значення та варіабельність виконання оптимізованої ОЗ newCO(…)

$\bar{x}$	*MED*	*S*	*W*
0,04259	0,04246	0,00009	0,00213
0,10027	0,10032	0,00012	0,00122
0,15701	0,15693	0,00023	0,00146
0,15969	0,15982	0,00016	0,00097
0,19694	0,19686	0,00024	0,00123
0,39356	0,39339	0,00017	0,00044
0,47034	0,47024	0,00040	0,00085
0,34871	0,34861	0,00027	0,00077
0,39878	0,39856	0,00046	0,00115
0,47659	0,47638	0,00065	0,00136
0,53202	0,53206	0,00028	0,00052
0,61070	0,61046	0,00060	0,00097
0,69725	0,69719	0,00070	0,00100
0,79887	0,79904	0,00028	0,00035

В таблицях (табл. 1, 2) зведено результати підрахунку типових значень (середнє та медіана) та варіабельності (середньоквадратичне

відхилення та W –коефіцієнт варіації Пірсона) відповідних вибірок. З аналізу таблиць, зважаючи на малу варіабельність спостережень (малі значення як вибіркового середньоквадратичного відхилення, так і коефіцієнта варіації Пірсона), порівняння швидкодії двох типів оптимізації операції згортки можна проводити шляхом співставлення типових значень. А зважаючи на те, що розподіли мають деяку відмінність від нормального, далі пропонується аналізувати медіани відповідних масивів спостережень.

Таблиця 3.

Оцінки медіан часу виконання згортки та відповідні різниці значень

M	MED_b [Sec]	MED_o [Sec]	MED_b - MED_o [Sec]	$\frac{MED\_b}{MED\_o}$
2	0,09478	0,04246	0,05232	2,2322
3	0,17575	0,10032	0,07543	1,7519
4	0,28521	0,15693	0,12828	1,8174
5	0,43033	0,15982	0,27051	2,6926
6	0,60632	0,19686	0,40946	3,08
7	0,81541	0,39339	0,42202	2,0728
8	1,82405	0,47024	1,35381	3,879
9	1,77476	0,34861	1,42615	5,091
10	1,86961	0,39856	1,47105	4,6909
11	1,87164	0,47638	1,39526	3,9289
12	1,87769	0,53206	1,34563	3,5291
13	1,88958	0,61046	1,27912	3,0953
14	1,88689	0,69725	1,18964	2,7062
15	1,89551	0,79904	1,09647	2,3722

В таблиці (табл.3) зведено медіанні оцінки часу виконання базової (MED_b) та оптимізованої (MED_o) згортки, а також різниці та відношення зазначених оцінок, відповідно до кожного з розмірів маски фільтру. Проаналізуємо швидкість зростання оцінок медіан для базової та оптимізованої згорток. Графік залежності оцінки медіани оптимізованої згортки від лінійного розміру маски фільтру згортки (рис. 1a) демонструє стале лінійне зростання:

$$MED\_o = 0,0536 \cdot l - 0,071 \tag{9}$$

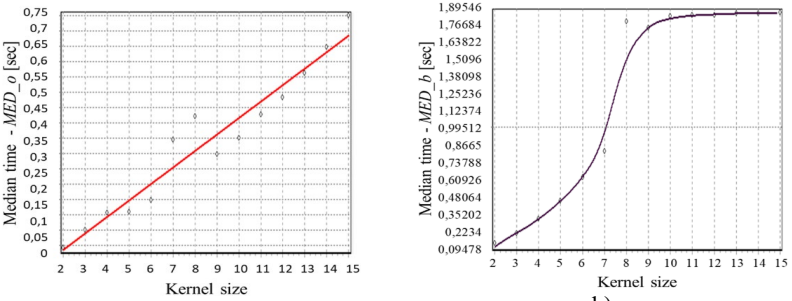
де l – лінійний розмір маски ($l = 2..15$).

Оцінка залежності медіани базової згортки від лінійного розміру маски фільтру згортки (рис. 1b) показує різну швидкість зростання, що є причиною неоднорідності різниці MED_b – MED_o. Як бачимо (рис. 1a), швидкість зростання для масок від 2x2 то 7x7 є лінійною з кутовим коефіцієнтом 0,144 (рис. 2a):

$$MED\_b = 0,144 \cdot l - 0,2467,\ l = 2..7 \tag{5}$$

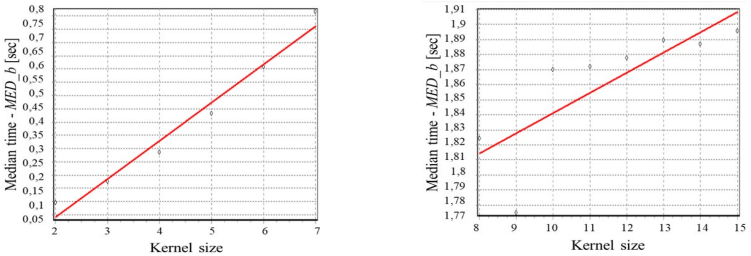
на відміну від масок 8x8 та більше (рис. 2b), де кутовий коефіцієнт зростання практично на порядок менший (0,0134):

$$MED\_b = 0,0134 \cdot l + 1,707,\quad l = 8..15. \tag{6}$$



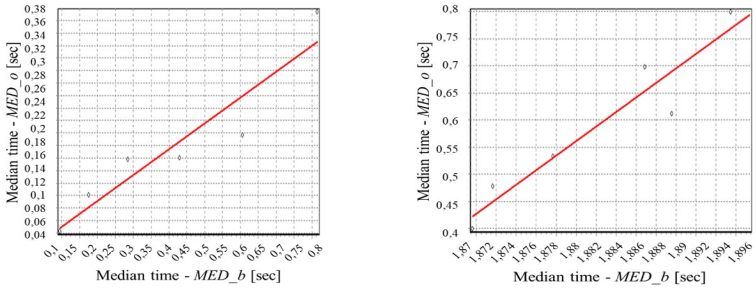
a) b)

Рис. 1. Оцінка залежності медіани від лінійного розміру маски фільтру: А) оптимізована згортка; Б) базова згортка.



a) b)

Рис. 2. Оцінка залежності медіани від лінійного розміру маски фільтру для базової згортки: А) розмір від 2x2 до 7x7; Б) розмір від 8x8 до 15x15



a) b)

Рис. 3. Оцінка лінійної залежності медіани часу оптимізованої згортки *MED_o* від базової *MED_b*: А) згортки від 2x2 до 7x7; Б) згортки від 9x9 до 15x15

Проведемо дослідження залежності оцінки медіани часу оптимізованої згортки *MED_o* від базової *MED_b.* Для розмірів згорток від 2x2

до 7x7 (рис. 3а) медіана часу виконання оптимізованої згортки лінійно залежить від часу виконання базової так:

$$MED\_o = 0,4134 \cdot MED\_b + 0,0091 \cdot \quad (7)$$

Тож, для масок такого розміру в середньому медіані час оптимізованої згортки становить майже 41% від базової.

Проте, для масок від 8x8 до 15x15 (рис. 3) маємо зовсім іншу тенденцію – перевага часу виконання оптимізованої згортки, у порівнянні з оптимізованою швидко зменшується (кутовий коефіцієнт залежності 13,437):

$$MED\_o = 13,437 \cdot MED\_b - 24,701 \cdot \quad (8)$$

З урахуванням результатів, а також залежностей (1) та (2), не складно показати, що вже починаючи з розмірів масок згорток більше, ніж 40x40 переваги в застосуванні оптимізованої згортки, у порівнянні з базовою фактично не буде і потрібно шукати нові рішення для такого випадку. Проте, для інших, менших розмірів згорток, авторами експериментально доведено, що використання запропонованої оптимізації є більш виправданим, ніж базовий підхід.

Висновок. Показано, що можна створити такі МЗ що задовольняють (3-4). Завдяки цьому було використано 16-bit операції SIMD NEON64. Також слід врахувати що досягнути пришвидшення вдалося завдяки одночасним завантаженням 48 (32+16) байт зображення для кожного рядка ЦЗ та 16 байт МЗ. Було виконано ряд вимірювань, які показують перевагу представленого підходу порівняно з функцією cv::filter2D(...) із бібліотеки OpenCV для ядер розміром менше 40x40. Оскільки поточні дослідження були обмежені однопоточними програмами та використанням одного методу для всіх розмірів зображень, ми припускаємо, що можна провести більше досліджень у напрямку розпаралелювання та використання різних методів для різних розмірів зображень і ядра. Крім того, ми очікуємо, що оптимізація циклів може призвести до ще більшого прискорення та отримання переваги для ядер розміром більше 40x40, але це залишилося для майбутніх досліджень [1].

Список літератури

1. Shevchenko, Andrii, Pylyp Prystavka, and Vitalii Tymchyshyn. "Research on Possible Convolution Operation Speed Enhancement via AArch64 SIMD." In *International Conference on Computer Science, Engineering and Education Applications*, pp. 61-75. Springer, Cham, 2022.
2. Shevchenko, Andrii, and Vitaly Tymchyshyn. A SIMD-based Approach to the Enhancement of Convolution Operation Performance[J], CMiGIN, 2019: 447-458
3. “Image Filtering,” Dec. 18 2021. Accessed on: Jan. 20 2022 [Online]. Ava ilable: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html#ga27c0497 95ce870216ddfb366086b5a04.
4. “Arm Compute Library,” Dec. 18 2021. Accessed on: Jan. 20 2022 [Onlin e]. Available: https://developer.arm.com/ip-products/processors/machine-learning/compute-library