Май 2024

Оценка вычислительной эффективности алгоритмов прямой свертки в нейронных сетях на центральных процессорах архитектуры ARM

Научный руководитель:
Лимонова Е.Е., кандидат технических наук

Автор:
Левин Л.И

Актуальность

Сверточные нейронные сети (CNN) представляют собой класс нейронных сетей, специально разработанных для выделения признаков, в первую очередь - на изображениях.

1. **Применение:**
   - **Компьютерное зрение**: для распознавания объектов, обработки изображений и даже создания искусственных фотографий.
   - **Медицинская диагностика**: для анализа медицинских изображений, диагностики болезней и предсказания результатов лечения.
   - **Обработка естественного языка**: для распознавания речи, машинного перевода и анализа текста.
2. **Операция свертки:**
   - Операция свертки в CNN является ключевым и наиболее вычислительно затратным компонентом, позволяющим сети автоматически извлекать пространственные и временные признаки из входных данных.
3. **Необходимость в оптимизации:**
   - В связи с вычислительными требованиями CNN, особенно при работе в режиме реального времени, вычислительно-эффективная реализация стала приоритетной задачей для улучшения производительности моделей.

Введение

Исследование развивает алгоритм из статьи[1] по анализу и оптимизации прямой свертки на многоядерных процессорах. Выделим главные плюсы и минусы данной статьи.

Плюсы

- Предложены различные способы оптимизации прямой свертки.
- Предложена эффективная методика подбора оптимизаций в зависимости от процессора.
- Изучена прямая свертка на процессорах архитектуры Intel и AMD.

Минусы

- Не изучена прямая свертка на процессорах архитектуры ARM.

1 - Analysis and Optimization of Direct Convolution Execution on Multi-Core Processors // Mannino M, Peccerillo B, Mondelli A, Bartolini S.

План

Цель работы:

- Оценить вычислительную эффективность алгоритма прямой свертки в нейронных сетях на центральных процессорах архитектуры ARM

Задачи:

- Реализовать прямую свертку на процессоре архитектуры ARM.
- Внедрить различные способы оптимизации и выявить наиболее эффективные.
- Сравнить время работы полученного алгоритма прямой свертки с современным алгоритмом im2col[2] и его аналогами.

2 - p-im2col: Simple Yet Efficient Convolution Algorithm With Flexibly Controlled Memory Overhead // A.V. Trusov, E.E. Limonova, D.P. Nikolaev, V.V. Arlazarov.

План оптимизации времени работы прямой свертки

1

Loop order

Выбор правильного порядка циклов может существенно влиять на производительность программы из-за особенностей работы с кэш-памятью и кэширования.

2

Loop tiling

Идея "tiling" заключается в разбиении обрабатываемого массива на более мелкие блоки,

3

Simd-vectorization

Технология, которая позволяет одной инструкции выполнять операции над несколькими элементами данных одновременно.

Данные для исследования

Layer ID	Input	Kernel
0	(227, 227, 3)	(11, 11, 3, 96)
1	(230, 230, 3)	(7, 7, 3, 64)
2	(226, 226, 3)	(3, 3, 3, 64)
3	(31, 31, 96)	(5, 5, 96, 256)
4	(58, 58, 64)	(3, 3, 64, 64)
5	(58, 58, 64)	(3, 3, 64, 128)
6	(58, 58, 128)	(3, 3, 128, 256)
7	(30, 30, 128)	(3, 3, 128, 128)
8	(30, 30, 256)	(3, 3, 256, 512)
9	(16, 16, 512)	(3, 3, 512, 512)
10	(15, 15, 384)	(3, 3, 384, 256)
11	(9, 9, 512)	(3, 3, 512, 512)

Исследование tiling на Apple M1

Layer ID	Loop order №1		Loop order №2		Loop order №3	
	Speedup	T.D.	Speedup	T.D.	Speedup	T.D.
0	1	Wo	1,05	Ho	0.98	Wo
1	1,18	Wo	1	Ho	0.96	Wo
2	1,18	Ci	1	Ho	0.94	Ci
3	1,02	Wo + Ci	1,02	Ho	0.93	Ci
4	1,18	Wo	1,08	Ho	1,03	Ci
5	1,01	Wo + Ci	0,99	Ho	1	Ci
6	1,02	Wo	1,08	Wo + Ho	1,01	Ci
7	1	Wo	1	Wo + Ho	1,01	Wo
8	1	Wo	0,96	Wo	1,12	Wo + Ci
9	1	Wo	1,02	Wo	1,14	Wo + Ci
10	1,02	Wo	1	Wo + Ho	1,07	Wo + Ci
11	1,05	Wo	1	Ho	1,1	Wo + Ci

- Экспериментально был найден наилучший размер tiling = (8, 32, 32).
- Максимальное ускорение loop_order1 составляет 18%
- Максимальное ускорение loop_order2 составляет 8%
- Максимальное ускорение loop_order3 составляет 14%

Исследование simd-инструкций с tiling на Apple M1

Layer ID	Simd Loop order №1		Simd Loop order №2		Simd Loop order №3	
	Speedup	T.D.	Speedup	T.D.	Speedup	T.D.
0	1	Ci	1,07	Ho	1	Ci
1	1,01	Ci	1,15	Ho	1	Ci
2	1,03	Ci	1,02	Ho	0,99	Ci
3	1,42	Wo + Ci	1,51	Wo	1,3	Wo
4	0.82	Ci	0,67	Ho	1,14	Wo + Ci
5	0.99	Ci	0,99	Ho	1	Wo
6	0,99	Wo + Ci	1,05	Ho	1,03	Ci
7	1	Wo	1,04	Ho	0,99	Wo
8	1,02	Ci	1	Wo + Ho	1,12	Wo + Ci
9	1	Ci	1,07	Wo	1,1	Ci
10	1	Wo	1,02	Wo	1,1	Ci
11	1,03	Ci	1,12	Wo	1,1	Ci

- Максимальное ускорение simd_loop_order1 составляет 42%
- Максимальное ускорение simd_loop_order2 составляет 51%
- Максимальное ускорение simd_loop_order3 составляет 14%

Исследование tiling на Agx Orin

Layer ID	Loop order №1		Loop order №2		Loop order №3	
	Speedup	T.D.	Speedup	T.D.	Speedup	T.D.
0	1,34	Wo + Ci	1,49	Ho	1,13	Wo + Ci
1	1,06	Wo + Ci	1,35	Ho	1,04	Ci
2	1,07	Wo + Ci	1,27	Ho	1,04	Ci
3	1,06	Wo + Ci	1	Wo + Ho	1,24	Ci
4	1,03	Wo + Ci	0,99	Ho	1,03	Ci
5	1,06	Wo + Ci	1,01	Ho	1,06	Ci
6	1,1	Wo + Ci	0.99	Ho	1,23	Ci
7	1,07	Wo + Ci	1	Wo	1,09	Ci
8	1,07	Wo + Ci	1,01	Wo + Ho	1,21	Wo + Ci
9	1,25	Wo + Ci	1	Wo	1,34	Ci
10	1,07	Wo + Ci	1	Wo	1,23	Wo + Ci
11	1,13	Wo + Ci	1	Wo	1,36	Ci

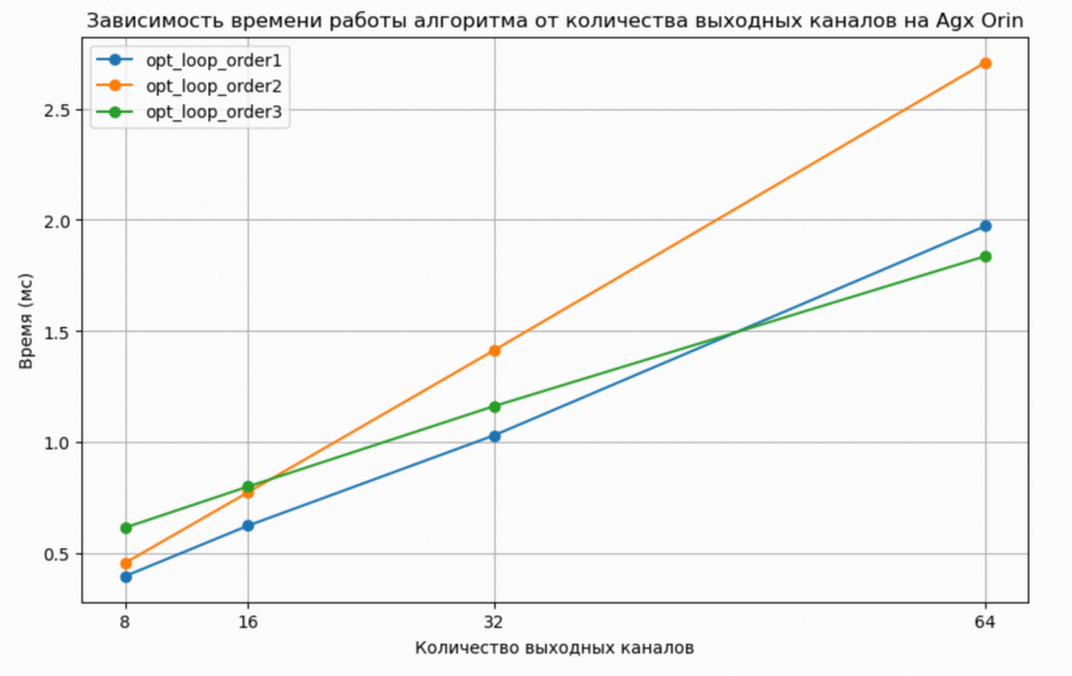
- Экспериментально был найден наилучший размер tiling = (8, 32, 32).
- Максимальное ускорение loop_order1 составляет 34%
- Максимальное ускорение loop_order2 составляет 49%
- Максимальное ускорение loop_order3 составляет 36%

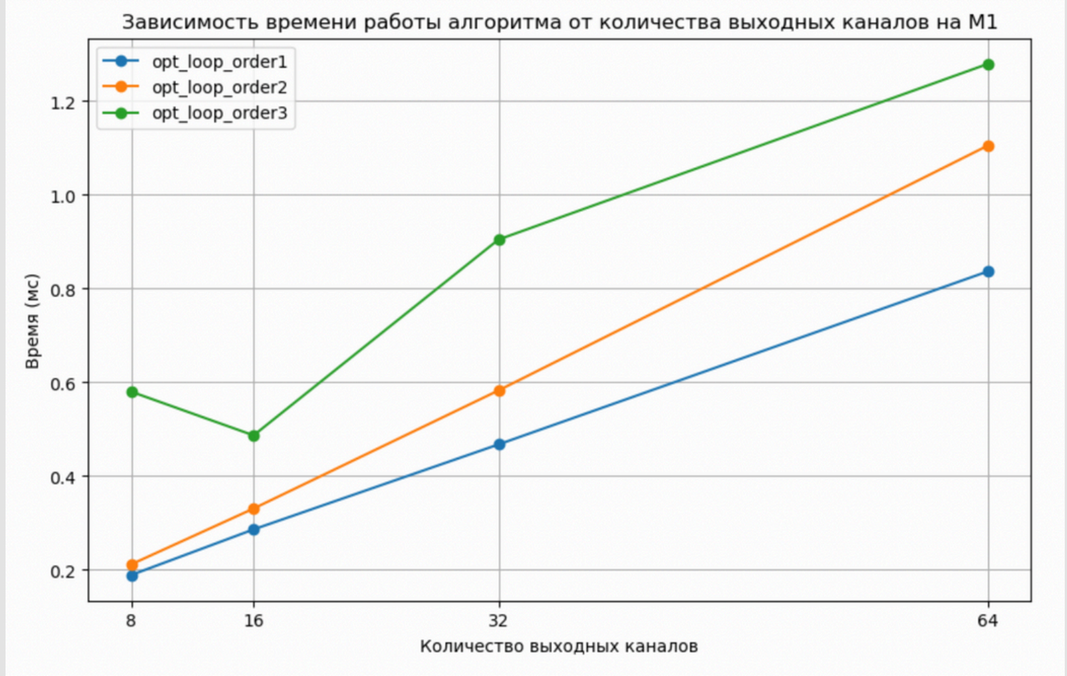
Исследование simd-инструкций с tiling на Agx Orin

Layer ID	Simd Loop order №1		Simd Loop order №2		Simd Loop order №3	
	Speedup	T.D.	Speedup	T.D.	Speedup	T.D.
0	1,38	Wo + Ci	1,45	Ho	1,12	Wo + Ci
1	1,03	Wo + Ci	1,38	Ho	1,02	Wo + Ci
2	1,04	Wo + Ci	1,25	Ho	1,01	Ci
3	1	Wo	1,01	Wo	1,24	Wo + Ci
4	1	Wo	0,99	Ho	1,03	Ci
5	1,02	Wo	1,01	Ho	1,02	Ci
6	1,1	Wo	0.99	Ho	1,21	Ci
7	1	Wo	1	Ho	1,06	Ci
8	1,03	Ci	1	Wo	1,2	Ci
9	1,16	Ci	1	Wo	1,33	Ci
10	1	Ci	1	Wo	1,22	Wo + Ci
11	1,06	Ci	1	Ho	1,37	Ci

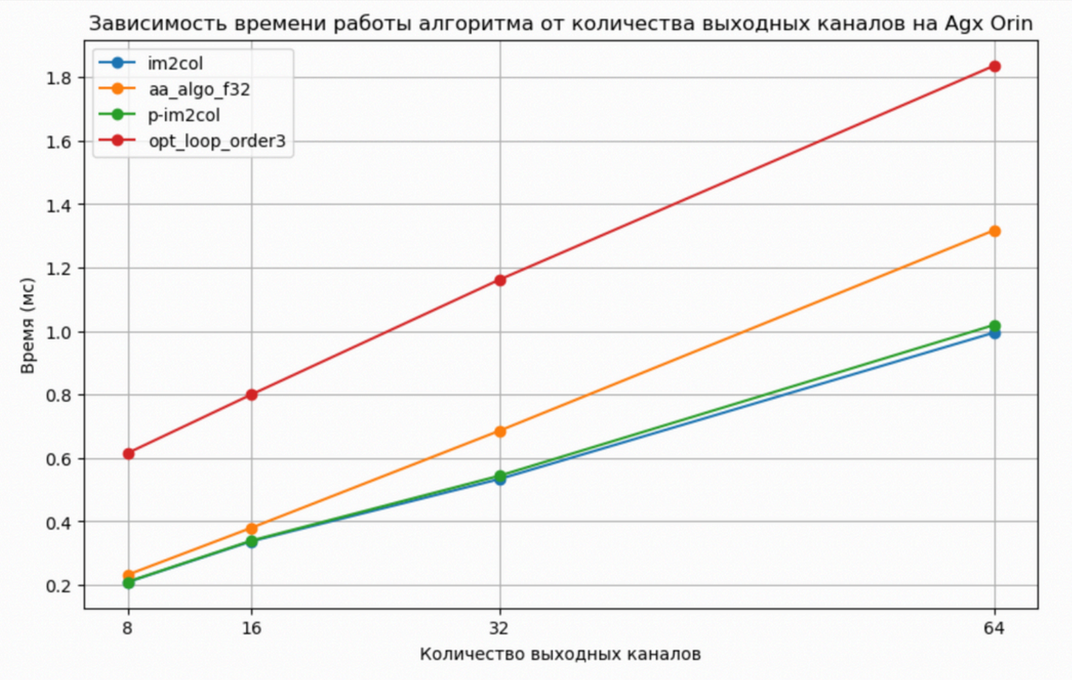
- Максимальное ускорение simd_loop_order1 составляет 38%
- Максимальное ускорение simd_loop_order2 составляет 45%
- Максимальное ускорение simd_loop_order3 составляет 37%

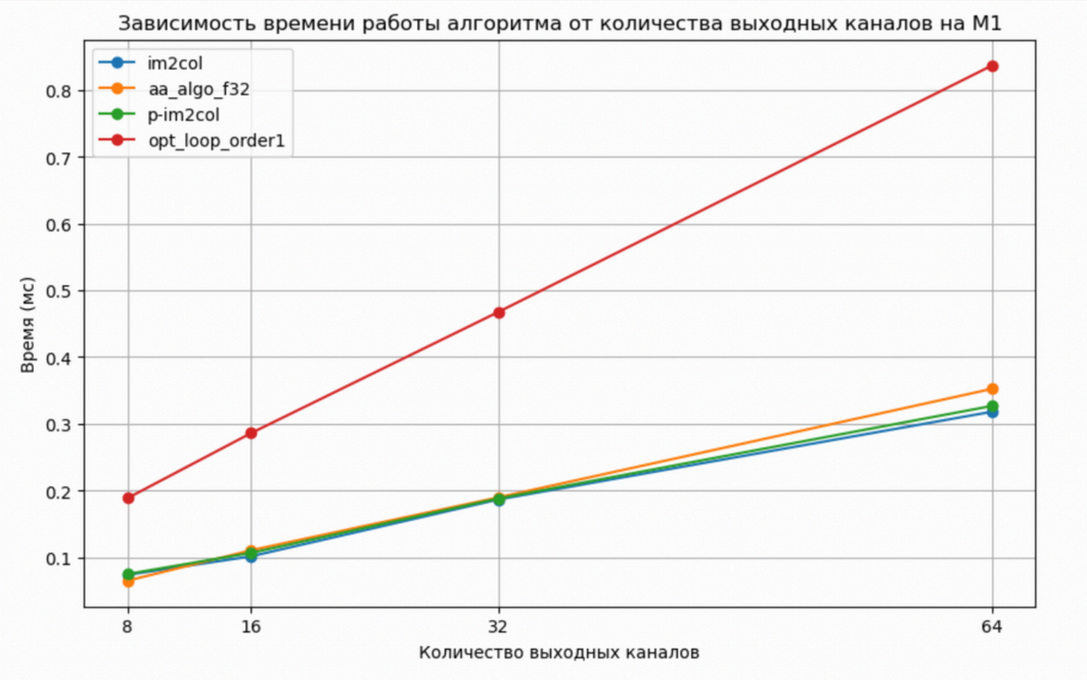
Сравнение разных версий прямой свертки





Сравнение прямой свертки с im2col и его аналогами





opt_loop_order - наилучшая версия прямой свертки с использованием simd и tiling

Выводы

- На процессоре Agx Orin при размерах выходных каналов больше 64 по эффективности лидирует loop_order3.
- На процессоре M1 при любых выходных каналов по эффективности лидирует loop_order1.
- Tiling ускоряют прямую свертку не более, чем на 50%
- На процессорах ARM алгоритм Im2col быстрее, чем оптимизированная прямая свертка.