Р. П. Сухоцкий

ИСПОЛЬЗОВАНИЕ ВЕКТОРНЫХ ИНСТРУКЦИЙ ПРОЦЕССОРА ДЛЯ ОПТИМИЗАЦИИ ОПЕРАЦИИ СВЕРТКИ

Исследована операция свертки для растровых изображений. Описано применение векторных инструкций процессора для ее оптимизации. Приведены результаты векторизации свертки на языке C#.

Введение

Одной из ключевых операций сверточных нейронных сетей (CNN) является свертка [1]. Свертка – это операция над локальной окрестностью изображения, где каждый результирующий пиксель представляет собой взвешенную сумму исходных пикселей. Ввиду большого количества вызовов данной функции, при работе сверточной нейронной сети, актуальным является сокращение ресурсоемкости данной операции. Для этого предлагается использовать векторные инструкции процессора. Такой подход позволит за один такт рассчитывать сразу несколько результирующих пикселей, что даст существенный прирост к скорости выполнения операции.

I. SIMD инструкции

SIMD (от англ. single instruction, multiple data – одиночный поток команд, множественный поток данных) – принцип организации компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных [2]. В современных процессорах такой принцип реализуется благодаря SIMD-инструкциям процессора (аппаратная поддержка) – командам в которых операндами могут выступать упорядоченные массивы данных (векторы).

Для архитектуры x86 существует ряд популярных SIMD-расширений (MMX, SSE, AVX). В зависимости от поддержки этих расширений пользователю доступны векторные регистры размером от 64 до 512 бит.

II. Векторизация операции свертки

Используя SIMD-инструкции в реализации свертки можно рассчитать значение не одного пикселя, как при стандартной реализации, а сразу вектора пикселей. Данное решение позволит получить теоретический прирост производительности кратный размеру используемого вектора.

Для оценки векторизации были написаны 4 функции: SimpleConv – простая реализация операции свертки, VectorConv – с использованием SIMD, _D – с двумерным массивом и _J – с массивом массивов. Тестовая программа была написана на языке C# с использованием специального функционала (класса Vector) для работы с новым компилятором (RyuJIT), автоматически генерирующего SIMD-инструкции с учетом аппаратной поддержки. Тестирование проводилось на машине с поддержкой SSE 4.2 и AVX2 (результаты приведены в табл.1).

Таблица 1 – Ресурсоемкость функций свертки

Функция свертки	Среднее время выполнения (мс)
SimpleConv_D	104.9507
VectorConv_D	81.2913
SimpleConv_J	96.0608
VectorConv_J	16.5649

Данные в таблице показывают, что векторизованная операция свертки выполняется быстрее, несмотря на временные затраты связанные с организацией промежуточного буфера, требуемого для обработки двумерного массива с применением SIMD. Реализация без промежуточного буфера в среднем выполнялась в 6 раз быстрее при использовании вектора из 8 элементов. Расхождение с ожидаемым приростом объясняется недостаточно эффективной работой с памятью (промахами в кэше), что требует дополнительной оптимизации.

III. Выводы

Предлагаемый способ оптимизации существенно ускоряет выполнение операции свертки, что повышает производительность обработки изображений с использованием сверточных нейронных сетей. Кроме того, свертка является одной из базовых операций цифровой обработке сигналов на основе которой реализуются фильтры сглаживания, контрастирования, шумоподавление и др.

Список литературы

1. LeCun, Y. Convolutional Networks for Images, Speech, and Time-Series. The Handbook of Brain Theory and Neural Networks / Y. LeCun, Y. Bengio // MIT Press. – 1995. – Vol. 7, № 1. – P. 255–258.
2. Flynn, M. J. Very High–Speed Computing Systems / M. J. Flynn // Proceedings of the IEEE. – 1966. – Vol. 54. – P. 1901–1909.

Сухоцкий Роман Петрович, студент факультета электронно-информационных систем БрГТУ, romansuchockij@gmail.com.

Научный руководитель: Кузьмицкий Николай Николаевич, старший преподаватель кафедры электронных вычислительных машин и систем БрГТУ, knnbrest@yandex.ru.