Оптимизация сверточных сетей с помощью квантизации и OpenVINO при распознавании снимков багажа

Н.А. Андриянов
Финансовый университет при Правительстве Российской Федерации
Москва, Россия
naandriyanov@fa.ru

Дж. Папакостас
Международный греческий университет
Салоники, Греция
gpapak@cs.ihu.gr

***Аннотация*—Работа посвящена исследованию ускорения инференса нейронных сетей с помощью квантизации весов и применения инструментария Intel OpenVINO Toolkit. При этом в исследовании рассматриваются сверточные сети блочной архитектуры, обучаемые с нуля. Показано, что применение OpenVINO обеспечивает значительное ускорение без потери качества работы для таких сетей, в то время как квантизация приводит к существенным потерям качества.**

Ключевые слова— компьютерное зрение, инференс, оптимизация, сверточные нейронные сети, квантизация, OpenVINO.

1. Введение

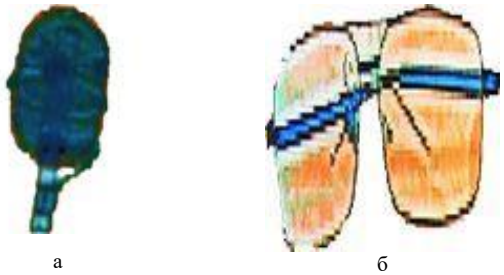
В последнее время с ростом сложности глубоких архитектур нейронных сетей, имеющих несколько миллиардов параметров [1], актуальным становится вопрос оптимизации работы таких моделей непосредственно при инференсе (логический вывод), т.е. при непосредственной работе сети. В частности, в задачах машинного зрения в ряде прикладных задач требуется вывод результатов практически в режиме реального времени.

Важной является задача распознавания и обнаружения объектов на изображениях [2]. Существует ряд подходов оптимизации, успешно применяемых в задаче ускорения работы глубоких сетей [3, 4], в том числе при обработке оптических изображений. В их числе прунинг, заключающийся в удалении весов и связей модели, квантизация весов и дистилляция, предназначенная для переноса знаний сети, например, с десктопного компьютера на мобильные устройства, не обладающие такими же большими вычислительными мощностями.

Другим решением является аппаратное ускорение или ускорение с учетом конкретных аппаратных платформ. К таким решениям относится ускоритель работы на процессорах Intel – OpenVINO Toolkit [5, 6]. Данный инструмент хорошо зарекомендовал себя при анализе оптических изображений. Однако применение таких алгоритмов при обработке изображений в других частотных диапазонах недостаточно изучено. В данной работе предлагается исследование ускорения инференса для рентгеновских снимков багажа и ручной клади на базе изображений аэропорта Баратаевка (г. Ульяновск). Задачи распознавания таких изображений более детально рассматривались в работе [7], а в настоящей работе основное внимание уделяется производительности сетей.

2. Обучение и Инференс

Рассмотрим задачу определения запрещенных к проносу предметов багажа и ручной клади. Очевидно, что важным показателем является именно полнота обнаружения таких предметов. При этом также важна и точность, поскольку при малой точности практически на каждый багаж будет требоваться ручная перепроверка. В исходной базе изображения были размечены под задачу распознавания. Задача детекции для снимков такого рода представляет дополнительные сложности. При сведении задачи к распознаванию удалось получить изображения отдельных предметов, которые были разделены на 2 класса. На рис. 1 представлены изображения запрещенного предмета (рис. 1а) и разрешенного предмета (рис. 1б).



а б

Рис. 1. Пример запрещенных (а) и разрешенных (б) объектов

В целом объем базы изображений составил 4000 изображений разрешенных объектов и 2000 – запрещенных. Для распознавания были обучены с нуля (с использованием библиотек Keras и TensorFlow) две сверточные сети. В первой, включающей 3 слоя свертки, использовались 128, 64 и 32 фильтра, оптимизация ADAM и 2000 эпох обучения, шаг обучения – 0,001. Вторая сеть состояла из 5 слоев свертки 256, 128, 64, 32 и 16 фильтров соответственно. Остальные параметры второй сети были выбраны, как и у первой. При этом размер обучающей выборки «разрешенных» изображений составил 2500 картинок, а «запрещенных» – 1200 картинок. Обучение происходило на видеокарте NVIDIA GeForce GTX 1060. Объем тестовой выборки составил 1500 и 800 изображений для «разрешенных» и «запрещенных» объектов соответственно. Размеры изображений 200 на 200 пикселей. Далее будем для простоты называть трехслойную сеть CNN-3, а пятислойную – CNN-5. В таблицах 1 и 2 представлены матрицы ошибок для CNN-3 и CNN-5.

Таблица I. МАТРИЦА ОШИБОК CNN-3

Истинный класс	**Прогнозируемый класс**		
	Разрешенный	***Запрещенный***	***Всего***
Разрешенный	1275	225 (FP)	1500
Запрещенный	168 (FN)	632 (TP)	800

Таблица II. МАТРИЦА ОШИБОК CNN-5

Истинный класс	**Прогнозируемый класс**		
	Разрешенный	***Запрещенный***	***Всего***
Разрешенный	1310	190 (FP)	1500
Запрещенный	127 (FN)	673 (TP)	800

Рассчитаем recall и precision для класса «запрещенных» предметов:

$$recall = \frac{TP}{TP + FN},\ precision = \frac{TP}{TP + FP} \quad (1)$$

где TP – количество «запрещенных» объектов, спрогнозированных, как «запрещенные»; FN – количество «запрещенных» объектов, спрогнозированных, как «разрешенные». Соответственно FP показывает количество «разрешенных» объектов, спрогнозированных, как «запрещенные».

В соответствии с выражением (1) $recall(3) = 0{,}79$, $recall(5) = 0{,}84$, $precision(3) = 0{,}74$, $precision(5) = 0{,}78$.

Инференс обученных нейронных сетей CNN-3 и CNN-5 будем выполнять на CPU Intel Core i7-8750. Выполним оценку показателя FPS (число кадров, обрабатываемых в секунду) на всей тестовой выборке. В связи с тем, что разные изображения в разных условиях могут обрабатываться за разное время, оценку производительности представим в виде $m_{FPS} \pm \sigma_{FPS}$, где m_{FPS} – среднее значение FPS по всем изображениям, а σ_{FPS} – стандартное отклонение по всем изображениям.

Далее выполним квантизацию весов до размеров INT8 (изначально сеть обучается в типе FP32), т.е. веса модели будут принимать только целочисленные значения в диапазоне [-128; 127]. Введем обозначения CNN-3q и CNN-5q соответственно для трехслойной и пятислойной сетей.

Наконец, реализуем Inference Engine с помощью инструмента Intel OpenVINO Toolkit. Следует отметить, что при первом запуске OpenVINO происходят конфигурационные настройки, другими словами, «разогрев нейросети». В связи с этим измерения производительности выполнялись уже после выполненной конфигурации под оптимальную работу на процессоре Intel. Оптимизированные с помощью OpenVINO сети из 3 и 5 слоев соответственно обозначим, как CNN-3ov и CNN-5ov.

В таблице 3 представлена оценка производительности сети, а также получаемые в условиях оптимизации характеристики полноты (recall) для тестовой выборки.

Анализ представленных результатов показывает, что использование OpenVINO целесообразно как для ускорения сети, так и для обеспечения сохранения метрик эффективности.

Таблица III. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Архитектура	***FPS***	***Recall***
CNN-3	$2{,}23 \pm 0{,}09$	0,79
CNN-5	$1{,}98 \pm 0{,}12$	0,84
CNN-3q	$9{,}54 \pm 0{,}11$	0,68
CNN-5q	$7{,}65 \pm 0{,}10$	0,72
CNN-3ov	$29{,}03 \pm 5{,}96$	0,79
CNN-5ov	$23{,}69 \pm 6{,}74$	0,84

3. ЗАКЛЮЧЕНИЕ

В работе предложено использовать ускорители для инференса нейронных сетей. Показано, что с помощью квантизации выполняется ускорение примерно в 3,5–4 раза. При этом полнота распознавания запрещенных объектов падает примерно на 10%. В случае использования инструментария Intel OpenVINO Toolkit удается сохранить заявленные характеристики сети, при этом выигрыш для производительности в среднем составляет порядка 11-13 раз. Однако следует учитывать, что применение OpenVINO обеспечивает время инференса с большим разбросом.

БЛАГОДАРНОСТИ

Исследование выполнено при поддержке Совета по грантам Президента Российской Федерации в рамках реализации Проекта по Стипендии Президента РФ молодым ученым и аспирантам № СП-3738.2022.5 и частично при поддержке гранта РФФИ № 19-29-09048.

ЛИТЕРАТУРА

[1] Romero, A. DeepMind Is Now the Undisputed Leader in Language AI with Gopher (280B) [Electronic resource]. – Access mode: https://towardsdatascience.com/deepmind-is-now-the-undisputed-leader-in-language-ai-with-gopher-280b-79363106011f (21.02.2022).

[2] Андриянов, Н.А. Обнаружение объектов на изображении: от критериев Байеса и Неймана–Пирсона к детекторам на базе нейронных сетей EfficientDet / Н.А. Андриянов, В.Е. Дементьев, А.Г. Ташлинский // Компьютерная оптика. – 2022. – Т. 46, № 1. – С. 139-159. DOI: 10.18287/2412-6179-CO-922.

[3] Kim, J. PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation / J. Kim, S. Chang, N. Kwak // ArXiv preprint: 2106.14681, 2021.

[4] Zhou, Y. Adaptive quantization for deep neural network / Y. Zhou, S.M. Moosavi-Dezfooli, N.M. Cheung, P. Frossard // ArXiv preprint: 1712.01048, 2017.

[5] Andriyanov, N.A. Analysis of the Acceleration of Neural Networks Inference on Intel Processors Based on OpenVINO Toolkit // Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), 2020. – P. 1-5. DOI: 10.1109/SYNCHROINFO49631.2020.9166067.

[6] Zunin, V.V. Intel OpenVINO Toolkit for Computer Vision: Object Detection and Semantic Segmentation // International Automation Conference (RusAutoCon), 2021. – P. 847-851.

[7] Andriyanov, N. Automatic x-ray image analysis for aviation security within limited computing resources / N. Andriyanov, Al. Volkov, An. Volkov, A. Gladkikh, S. Danilov // IOP Conference Series: Materials Science and Engineering, 2020. – Vol. 862. – P. 052009. DOI: 10.1088/1757-899X/862/5/052009.