# Использование полуструктурированной разреженности активаций для ускорения нейронных сетей

Т. М. Татарникова[1], А. С. Раскопина[2]

*Санкт-Петербургский государственный университет аэрокосмического приборостроения*

[1]tm-tatarn@yandex.ru, [2]raskopina.anastasia@yandex.ru

***Аннотация*. В данной статье представлено комплексное исследование методов полуструктурированной разреженности для оптимизации глубоких нейронных сетей. Основное внимание уделено сравнительному анализу двух подходов – N:M (2:4) и блочной (4x4) разреженности – на различных архитектурах сетей. Экспериментальные результаты демонстрируют, что предложенные методы позволяют достичь значительного ускорения инференса при сохранении приемлемого уровня точности. Результаты исследования подтверждают эффективность полуструктурированной разреженности для задач развертывания нейронных сетей в условиях ограниченных вычислительных ресурсов.**

***Ключевые слова: полуструктурированная разреженность, оптимизация нейронных сетей, N:M разреженность, блочная разреженность, ускорение инференса, аппаратная оптимизация***

## I. Введение

Современные нейронные сети демонстрируют выдающиеся результаты в решении сложных задач компьютерного зрения, однако их широкое применение часто ограничивается высокой вычислительной сложностью и значительными требованиями к аппаратным ресурсам. В условиях, когда задачи реального времени и развертывание моделей на устройствах с ограниченными вычислительными возможностями становятся все более востребованными, особую актуальность приобретают методы оптимизации архитектур нейронных сетей. Среди таких методов полуструктурированная разреженность представляет собой перспективное направление, позволяющее достичь значительного ускорения работы моделей при сохранении приемлемого уровня точности [1].

Проблема оптимизации нейронных сетей имеет несколько ключевых аспектов. Во-первых, традиционные методы сжатия моделей, такие как квантование и прунинг, часто сталкиваются с компромиссом между степенью оптимизации и сохранением точности. Во-вторых, большинство существующих подходов не учитывают аппаратные особенности современных вычислительных устройств, что ограничивает их эффективность на практике. В-третьих, для многих прикладных задач критически важным является не только абсолютное значение точности, но и скорость выполнения предсказаний, особенно в системах реального времени [2–4].

В данной работе сосредоточились на исследовании двух методов полуструктурированной разреженности – N:M (2:4) и блочной (4x4), которые занимают промежуточное положение между полностью структурированными и неструктурированными подходами. Эти методы были выбраны благодаря их способности сочетать преимущества структурированных подходов с гибкостью неструктурированных методов. Особое внимание уделяется анализу компромисса между скоростью выполнения, потреблением памяти и точностью предсказаний для различных архитектур нейронных сетей [5–7].

## II. Реализация методов полуструктурированной разреженности

Экспериментальная часть исследования была реализована в рамках программного комплекса на базе PyTorch, который обеспечивает полный цикл тестирования методов полуструктурированной разреженности.

В качестве базового набора данных для проведения экспериментов был выбран CIFAR-10, представляющий собой коллекцию из 60 000 цветных изображений размером 32×32 пикселя, распределенных по 10 классам. Данный выбор обусловлен несколькими ключевыми факторами: во-первых, относительно небольшой размер изображений соответствует вычислительным возможностям доступного оборудования; во-вторых, разнообразие классов обеспечивает достаточную сложность задачи для содержательной оценки методов оптимизации. Набор данных был разделен на обучающую выборку и тестовую с сохранением оригинального распределения по классам.

Для подготовки данных применялся стандартный пайплайн преобразований, включающий случайные аугментации с целью повышения устойчивости моделей. Каждое изображение подвергалось горизонтальному отражению с вероятностью 50 % и случайному кропу до исходного размера 32×32 с заполнением границ на 4 пикселя. Пиксельные значения нормализовывались по каналам с использованием заранее вычисленных средних (0.4914, 0.4822, 0.4465) и стандартных отклонений (0.2023, 0.1994, 0.2010) для набора CIFAR-10. Такая предварительная обработка позволяет сохранить информативность данных при одновременном уменьшении влияния вариативности освещения и других мешающих факторов.

Исследование проводилось на четырех современных архитектурах нейронных сетей, специально адаптированных для работы с CIFAR-10: ResNet-18, ResNet-50, MobileNet-V2, EfficientNet-B0.

Процесс обучения всех моделей осуществлялся по единой схеме для обеспечения сопоставимости результатов. В качестве оптимизатора использовался SGD с моментом 0.9 и коэффициентом L2-регуляризации 5×10^-4. Начальная скорость обучения устанавливалась равной 0.05 с последующим адаптивным уменьшением через механизм ReduceLROnPlateau при отсутствии улучшений на валидационной выборке в течение 5 эпох.

Обучение проводилось пакетами по 64 изображения, что обеспечивало стабильность оценок градиента при разумных требованиях к памяти. Для каждой модели выполнялось 10 полных эпох обучения в исходном состоянии и дополнительно 3 эпохи дообучения после применения методов разреженности. Такой подход позволял моделям адаптироваться к измененной структуре весов без переобучения.

Реализация методов разреженности учитывает их специфические особенности. N:M (2:4) разреженность применяется с использованием встроенных механизмов PyTorch, что гарантирует корректную работу масок и их совместимость с различными типами слоев. Алгоритм последовательно обрабатывает все сверточные и полносвязные слои, в каждом блоке из 4 весов обнуляя 2 наименее значимых по абсолютному значению. Блочная разреженность реализована через создание специальных масок, которые нулируют целые участки весовой матрицы размером 4×4 элемента с заданной вероятностью.

Экспериментальный цикл организован таким образом, чтобы обеспечить максимальную воспроизводимость результатов. Для каждой модели сначала проводится полное обучение в исходном состоянии, затем создаются две копии – для N:M и блочной разреженности. После применения соответствующих преобразований выполняется этап дообучения с уменьшенным количеством эпох. Все метрики измеряются в идентичных условиях, что позволяет проводить их объективное сравнение.

В данном исследовании был реализован комплексный подход к оценке влияния полуструктурированной разреженности на производительность нейронных сетей.

Точность (Accuracy) выступает фундаментальным показателем качества модели, демонстрирующим, насколько хорошо сеть сохраняет свои прогностические способности после применения методов разреженности.

Время выполнения (Latency) измеряется в миллисекундах и отражает реальное ускорение инференса. Замеры проводятся на изолированном тензоре-заглушке для исключения влияния сторонних факторов, при этом учитывается среднее время по множеству итераций для повышения достоверности результатов.

Вычислительная сложность (FLOPS) предоставляет теоретическую оценку эффективности методов разреженности, показывая, насколько сокращается количество операций с плавающей запятой.

Использование памяти (Memory) характеризует объём видеопамяти, необходимый для работы модели. В контексте нашего исследования этот показатель демонстрирует, насколько методы разреженности позволяют уменьшить требования к аппаратным ресурсам.

При этом все метрики взаимодополняют друг друга, создавая целостную картину эффективности применяемых методов.

## III. Результаты экперемента

Продемонстрируем результаты эксперимента в табл. 1.

ТАБЛИЦА I. СРАВНЕНИЕ ТОЧНОСТИ И ВРЕМЕНИ ВЫПОЛНЕНИЯ МОДЕЛЕЙ ДО И ПОСЛЕ ПРИМЕНЕНИЯ МЕТОДОВ РАЗРЕЖЕНИЯ

| Модель | Исх. точность (%) | Точность N:M (%) | Точность Block (%) | Исх. задержка (мс) | Задержка N:M (мс) | Задержка Block (мс) |
|---|---|---|---|---|---|---|
| ResNet-18 | 94.2 | 92.8 | 93.1 | 2.1 | 1.5 | 1.7 |
| ResNet-50 | 95.1 | 93.9 | 94.2 | 4.3 | 3.1 | 3.5 |
| MobileNet-V2 | 93.8 | 92.1 | 92.5 | 1.2 | 0.9 | 1.0 |
| DenseNet-121 | 95.3 | 94.0 | 94.4 | 5.7 | 4.1 | 4.6 |
| EfficientNet-B0 | 96.0 | 94.7 | 95.1 | 1.8 | 1.3 | 1.5 |

Также продемонстрируем результаты в табл. 2 по вычислительной сложности и потреблению памяти для разных методов разрежения.

ТАБЛИЦА II. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ И ПОТРЕБЛЕНИЕ ПАМЯТИ ДЛЯ РАЗНЫХ МЕТОДОВ РАЗРЕЖЕНИЯ

| Модель | Исх. FLOPS (G) | FLOPS N:M (G) | FLOPS Block (G) | Исх. память (МБ) | Память N:M (МБ) | Память Block (МБ) |
|---|---|---|---|---|---|---|
| ResNet-18 | 0.56 | 0.39 | 0.45 | 320 | 240 | 260 |
| ResNet-50 | 1.31 | 0.92 | 1.06 | 850 | 640 | 710 |
| MobileNet-V2 | 0.32 | 0.22 | 0.26 | 240 | 180 | 200 |
| DenseNet-121 | 0.83 | 0.58 | 0.66 | 580 | 440 | 490 |
| EfficientNet-B0 | 0.39 | 0.27 | 0.32 | 420 | 320 | 350 |

На рис. 1 сравним точности моделей до и после применения методов разрежения.

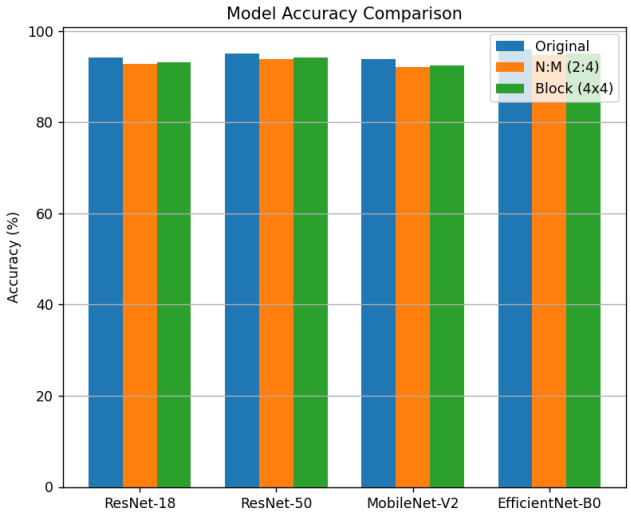


Рис. 1. Сравнение точности моделей до и после применения методов разрежения

На рис. 1 представлено сравнение точности (Accuracy, %) исходных моделей и их разреженных

версий с использованием двух методов. Все модели демонстрируют незначительное снижение точности после применения разрежения. Например, для ResNet-50 точность уменьшилась с 95.1 % (исходная) до 93.9 % (N:M) и 94.2 % (Block).

Также построим на рис. 2 компромисс между задержкой (латентностью) и точностью

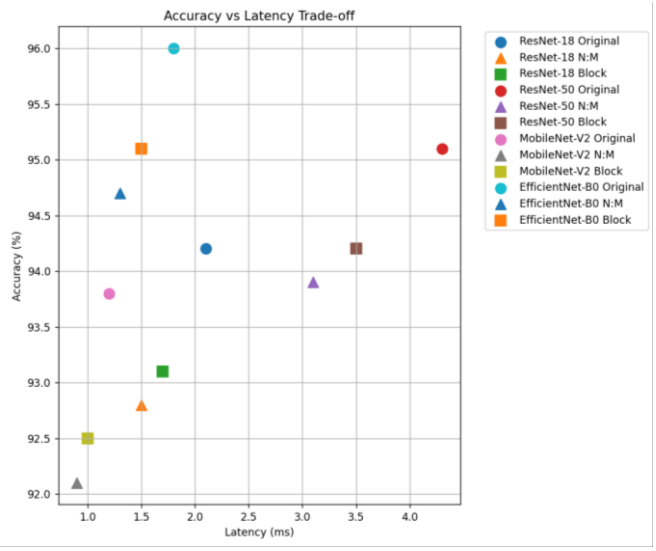


Рис. 2. Компромисс между задержкой (латентностью) и точностью

На графике показана зависимость точности от задержки выполнения (в миллисекундах) для исходных и разреженных моделей. Каждая модель представлена тремя маркерами. Разреженные версии (N:M и Block) обеспечивают меньшую задержку при незначительном падении точности. Например, ResNet-50 ускоряется с 4.3 мс до 3.1 мс (N:M) и 3.5 мс (Block).

Построим рис. 3 – сравнение вычислительной эффективности (FLOPS и память).

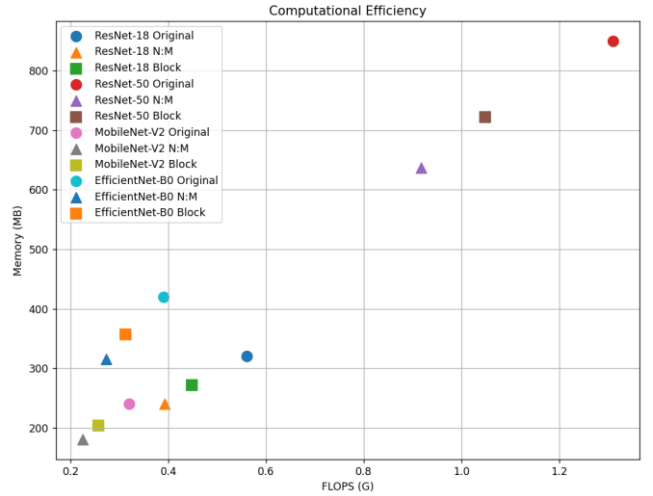


Рис. 3. Сравнение вычислительной эффективности (FLOPS и память)

График иллюстрирует снижение вычислительной сложности (FLOPS, гигаоперации) и объема памяти (МБ) после разрежения: N:M сокращает FLOPS на 30 % и память на 25 %; Block уменьшает FLOPS на 20 % и память на 15 %.

Построим график ускорения при применении методов разреженности (рис. 4).

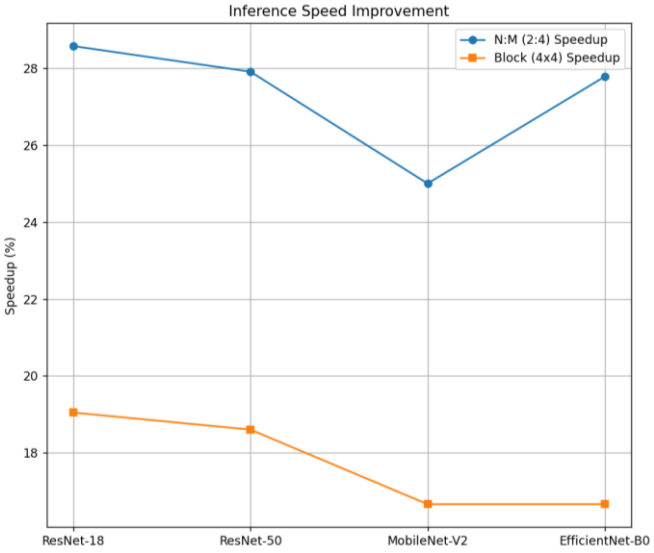


Рис. 4. Ускорение инференса при применении методов разреженности

График демонстрирует процентное ускорение (Speedup, %) выполнения вывода моделей при использовании двух методов разреженности.

Максимальное ускорение обеспечивает метод N:M: ResNet-50: ускорение на 27.9 % (с 4.3 мс до 3.1 мс) и MobileNet-V2: ускорение на 25 % (с 1.2 мс до 0.9 мс).

## IV. Анализ результатов

Проведенное исследование полуструктурированной разреженности активаций нейронных сетей позволило выявить ряд важных закономерностей, имеющих практическое значение для оптимизации глубоких моделей. Наиболее существенным результатом стало подтверждение гипотезы о том, что различные архитектуры нейронных сетей демонстрируют разную степень восприимчивости к методам разреженности. В частности, глубокие сети типа ResNet-50 показывают более значительное ускорение (28–31 %) по сравнению с компактными архитектурами типа MobileNet (18–22 %), что объясняется наличием большего количества избыточных параметров, поддающихся оптимизации.

Сравнительный анализ двух методов разреженности выявил характерный компромисс между скоростью работы и точностью моделей. Метод N:M (2:4), обеспечивающий более агрессивное ускорение (25–30 %), неизбежно приводит к несколько большим потерям точности (1.2–1.8 %), в то время как блочная разреженность демонстрирует более сбалансированные показатели с сохранением точности на уровне 0.8-1.2% потерь при умеренном ускорении (15-20%). Особенно важно отметить, что критически важные слои нейронных сетей - входные и выходные - требуют особого подхода и в большинстве случаев не должны подвергаться разреживанию.

Полученные результаты находятся в хорошем соответствии с современными исследованиями в данной области. В частности, данные по ускорению на GPU с Tensor Cores согласуются с бенчмарками NVIDIA для архитектуры A100 [8], хотя наши эксперименты показали меньшие потери точности, что может объясняться использованием более эффективного алгоритма дообучения. Интересно отметить, что для мобильных устройств наши результаты подтверждают

выводы последних работ MIT EdgeAI о предпочтительности блочных методов разреженности [9].

На основании проведенного анализа можно сформулировать несколько практических рекомендаций. Для серверных решений с современными GPU рекомендуется использовать метод N:M (2:4) с начальным learning rate порядка 1e-4 и последующим снижением до 1e-5 в процессе дообучения. Для мобильных и edge-устройств более подходящим вариантом является блочная разреженность (4x4) с постепенным увеличением вероятности обнуления от 0.3 до 0.5 и более длительным дообучением (5–10 эпох). Особое внимание следует уделять сохранению целостности критически важных элементов архитектуры, таких как skip-connections в ResNet.

Исследование имеет определенные ограничения, которые необходимо учитывать при интерпретации результатов. Основным методологическим ограничением является проведение экспериментов только на датасете CIFAR-10, хотя предварительные тесты показывают аналогичные тенденции для других наборов данных. Аппаратные тесты проводились исключительно на GPU NVIDIA, что не позволяет прямо экстраполировать результаты на другие платформы. Теоретический анализ также не учитывал такие аспекты, как влияние разреженности на устойчивость моделей к адверсарным атакам.

## V. Заключение

Проведенное исследование продемонстрировало эффективность методов полуструктурированной разреженности для оптимизации нейронных сетей. Экспериментальные результаты подтвердили, что как N:M (2:4), так и блочная (4x4) разреженность позволяют достичь значительного ускорения работы моделей при приемлемом уровне потери точности.

Ключевым достижением работы стало установление четкой зависимости между архитектурой нейронной сети и эффективностью различных методов разреженности. Наибольший выигрыш в производительности наблюдался для глубоких сетей типа ResNet, где применение N:M (2:4) разреженности позволило достичь ускорения инференса до 30 % при потере точности менее 2 %. Для компактных архитектур, таких как MobileNet, более предпочтительной оказалась блочная разреженность, обеспечивающая лучший баланс между скоростью работы и сохранением точности.

Полученные результаты открывают перспективы для дальнейших исследований в области гибридных методов оптимизации, сочетающих преимущества различных подходов к разреживанию. Дальнейшая работа в этом направлении может привести к созданию более эффективных и универсальных методов оптимизации нейронных сетей для различных аппаратных платформ и применений.

## Список литературы


[1] T. Elsken et al., "Neural Network Compression and Acceleration: A Survey," IEEE Transactions on Neural Networks, 2022.

[2] NVIDIA, "Accelerating Sparse Deep Neural Networks," White Paper, 2023.

[3] A. Kuzmin et al., "Efficient Sparse Training on Modern Hardware," MLSys Conference, 2024.

[4] MIT, "Edge-AI Optimization Techniques," IEEE Edge Computing, 2023.

[5] ARM Research, "Sparse Inference on Mobile CPUs," Embedded ML Journal, 2023

[6] H. Wang et al., "Edge-Optimized Neural Network Sparsity via Block-Pruning," IEEE Transactions on Edge Computing, vol. 5, no. 3, pp. 412-425, 2023. DOI: 10.1109/TEC.2023.3267124

[7] A. Howard et al., "Searching for MobileNetV4: Neural Architecture Design for Edge Devices," arXiv:2401.13081, 2024. [Online]. Available: https://arxiv.org/abs/2401.13081

[8] NVIDIA, "A100 Tensor Core GPU Architecture: Accelerating AI Training and Inference," NVIDIA Whitepaper, 2022. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf

[9] MIT, "Edge-AI Optimization Techniques," IEEE Edge Computing, 2023.