Программные системы и вычислительные методы
Правильная ссылка на статью:
Мамадаев И.М., Минитаева А.М. Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70658 EDN: LDXKKC URL: https://nbpublish.com/library_read_article.php?id=70658

Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS

Мамадаев Ибрагим Магомедович

Разработчик; ООО "Мэйл.Ру"

-, Россия, г. Москва, бул. Тверской, -

ibragim.m115@gmail.com

Минитаева Алина Мажитовна

кандидат технических наук

преподаватель; Информатика и Вычислительная Техника (ИУ-6); Московский Государственный Технический Университет им.Н.Э.Баумана

-, Россия, г. Москва, бул. Тверской, -

aminitaeva@mail.ru

Статья из рубрики "Операционные системы "

DOI:

10.7256/2454-0714.2024.2.70658

EDN:

LDXKKC

Дата направления статьи в редакцию:

05-05-2024

Дата публикации:

13-06-2024

Аннотация: На сегодняшний день мобильные устройства играют важную роль в

повседневной жизни каждого человека, и одной из ключевых технологий, приводящих к значительным преимуществам для мобильных приложений, является машинное обучение. Оптимизация алгоритмов машинного обучения для мобильных устройств является актуальной и важной задачей, она направлена на разработку и применение методов, которые позволят эффективно использовать ограниченные вычислительные ресурсы мобильных устройств. В работе рассматриваются различные способы оптимизации алгоритмов распознания изображений на мобильных устройствах, такие как квантизация и сжатие моделей, оптимизация изначальных вычислений. Помимо способов оптимизации самой модели машинного обучения рассматриваются также различные библиотеки и инструменты для применения данной технологии на мобильных устройствах. Каждый из описанных способов имеет свои преимущества и недостатки, в связи с чем в итогах работы предлагается использовать не только комбинацию описанных вариантов, а также дополнительный способ - параллелизацию процессов обработки изображений. В статье рассмотрены примеры конкретных инструментов и фреймворков, доступных для оптимизации производительности машинного обучения на iOS, проведены собственные эксперименты для проверки эффективности различных методов оптимизации. А также приведен анализ полученных результатов и сравнение производительности алгоритмов. Практическая значимость данной статьи заключается в следующем: - Улучшение производительности алгоритмов машинного обучения на мобильных устройствах iOS приведет к более эффективному использованию вычислительных ресурсов и повышению производительности системы, что является весьма важным в контексте ограниченных вычислительных мощностей и энергетических ресурсов мобильных устройств. - Оптимизация производительности машинного обучения на iOS-платформе способствует развитию более быстрых и отзывчивых приложений, что также улучшит пользовательский опыт и позволит разработчикам создавать новые и инновационные функции и возможности. - Расширение применимости машинного обучения на мобильных устройствах iOS открывает новые возможности для разработки приложений в различных областях, таких как распознавание образов, обработка естественного языка, анализ данных и другие.

Ключевые слова:

нейронная сеть, машинное обучение, мобильное устройство, iOS, распознавание изображений, оптимизация, ОС Apple, эффективность, производительность, параллелизация

Введение и актуальность

На сегодняшний день мобильные устройства играют важную роль в жизни каждого человека, так как предоставляют широкий спектр возможностей и сервисов, без которых многие уже и не могут представить свою повседневную жизнь. Одной из ключевых технологий, приводящих к значительным преимуществам мобильных приложений, является машинное обучение [1], оно уже применяется во многих ведущих приложениях на рынке, а крупные IT-компании конкурируют между собой, пытаясь привлечь на свою сторону большее количество клиентов. Однако, для их эффективного использования на мобильных устройствах, необходимо решить ряд непростых проблем.

Оптимизация алгоритмов машинного обучения для мобильных устройств является актуальной и важной задачей и направлена на разработку и применение методов, которые позволят эффективно использовать ограниченные вычислительные ресурсы

мобильных устройств, минимизировать энергопотребление и достигать высокой производительности при выполнении сложных задач машинного обучения. Проведение оптимизаций открывает новые возможности для развития приложений, таких как умные помощники и голосовые ассистенты, обработка изображений и видео в реальном времени, автоматическая классификация данных.

Вместе с ростом популярности и использования мобильных устройств, компании Apple [2], возникает и потребность в эффективной работе алгоритмов машинного обучения на ограниченных вычислительных ресурсах, малой памяти и недостаточного, ввиду размера, запаса аккумулятора.

Анализируя проблемы, связанные с производительностью машинного обучения на мобильных устройствах под управлением операционной системы iOS, можно выделить следующие аспекты:

- задержки в выполнении алгоритмов из-за их сложности,

- снижение отзывчивости пользовательского интерфейса из-за перегрузки вычислительных мощностей устройства,

- увеличение энергопотребления и как следствие повышение тепловыделения.

Перечисленные аспекты оказывают негативное влияние на пользовательский опыт и ставят перед разработчиками задачи по обеспечению высокой производительности приложения.

Целью данной работы является исследование и оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения на мобильных устройствах iOS. Основная задача состоит в исследовании существующих методов и техник оптимизации; проведении анализа производительности различных алгоритмов машинного обучения; оценке влияния различных факторов на производительность.

В статье рассмотрены примеры конкретных инструментов и фреймворков, доступных для оптимизации производительности машинного обучения на iOS, проведены собственные эксперименты для проверки эффективности различных методов оптимизации. А также приведен анализ полученных результатов и сравнение производительности алгоритмов.

Практическая значимость данной статьи заключается в следующем:

- Улучшение производительности алгоритмов машинного обучения на мобильных устройствах iOS приведет к более эффективному использованию вычислительных ресурсов и повышению производительности системы, что является весьма важным в контексте ограниченных вычислительных мощностей и энергетических ресурсов мобильных устройств.

- Оптимизация производительности машинного обучения на iOS-платформе способствует развитию более быстрых и отзывчивых приложений, что также улучшит пользовательский опыт и позволит разработчикам создавать новые и инновационные функции и возможности.

- Расширение применимости машинного обучения на мобильных устройствах iOS открывает новые возможности для разработки приложений в различных областях, таких как распознавание образов, обработка естественного языка, анализ данных и другие.

1 Обзор существующих решений

На мобильных устройствах iOS довольно часто используются всевозможные вариации алгоритмов машинного обучения для решения разнообразных задач [3]. Некоторые из них включают в себя алгоритмы классификации, регрессии, кластеризации, нейронные сети и глубокое обучение [4].

Яркими примерами таких моделей являются алгоритмы классификации, логистическая регрессия и метод опорных векторов, которые широко применяются для решения задач распознавания образов и классификации данных на мобильных устройствах.

Данные методы обладают относительно низкой сложностью и хорошо масштабируются для работы с большими объемами данных [5]. Алгоритмы регрессии, которые включают в себя линейную регрессию и метод наименьших квадратов, используются для предсказания численных значений на основе исходных данных. Такие алгоритмы широко применяются в задачах прогнозирования и анализа данных на мобильных устройствах. Кластеризация, в свою очередь, является методом группировки схожих объектов на основе их характеристик. Некоторые алгоритмы кластеризации, такие как k-средних и DBSCAN, используются для обработки данных на мобильных устройствах и поиска скрытых структур. Нейронные сети и глубокое обучение также являются одними из самых популярных алгоритмов машинного обучения на сегодняшний день, так как могут обрабатывать сложные данные, изображения, тексты, и при этом достигать высокой точности в задачах классификации, распознавания и генерации контента. Правильное использование методов оптимизации может значительно повысить эффективность работы алгоритмов машинного обучения на мобильных устройствах.

2 Квантизация моделей

Одним из ключевых методов оптимизации является квантизация моделей [6], она позволяет уменьшить размер модели и снизить требования к вычислительным ресурсам путем представления весов и активаций с меньшей точностью. Иными словами – квантование это процесс снижения точности весов путем округления, уменьшения точности. Наглядная иллюстрация данного процесса приведена на рисунке 1.

Рисунок 1 – Пример квантования веса 1 нейрона и снижение разрядности в 4 раза

Применение квантизации позволяет ускорить процесс вычислений и снизить использование памяти, несильно влияя на точность модели. Одним из главных преимуществ квантизации является сокращение размера модели, что в свою очередь приводит к уменьшению требований как к памяти устройства, так и непосредственно к самому ускорению вычислений. Кроме того, квантизация позволяет использовать

специализированные аппаратные ускорители, такие как «Neural Engine» [2] в чипах от компании Apple, предназначенных в том числе для эффективного выполнения операций с низкой точностью. Однако у квантизации есть весьма большой недостаток - она может также привести к потере точности модели, особенно при использовании уменьшении точности представления весов и активаций.

Однако этот весомый недостаток перекрывается следующим преимуществом-особенностью данного метода является то, что его можно использовать как во время обучения модели, так и после, что позволяет проводить подобные операции и после поставки приложения пользователям.

3 Сжатие моделей

Еще одним методом оптимизации является сжатие моделей, которое позволяет уменьшить размер модели, удаляя ненужные или избыточные параметры. Одним из видов сжатия моделей является «прунинг» [7] (отсечение или прореживание). Графики точности и производительности моделей в зависимости от процента прореживания таким способом приведены на рисунке 2.

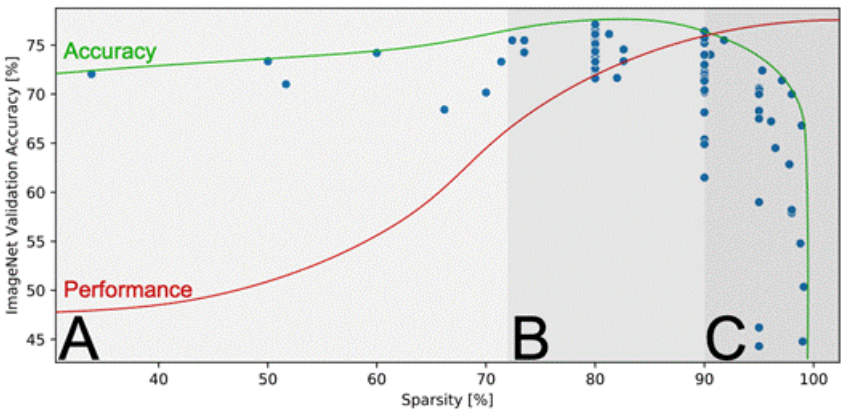

Рисунок 2 – Графики точности и производительности при прунинге (прореживании) моделей

Особенностью прунинга, в отличие от квантизации, является то, что данный процесс возможен только в уже предобученной модели.

Сжатие моделей подобным образом также имеет свои преимущества, оно позволяет уменьшить размер модели, что упрощает развертывание и ускоряет загрузку на устройствах. Оно также может снизить требования к памяти и вычислительным ресурсам. Однако, при сжатии моделей существует риск потери информации и точности модели. Некоторые методы сжатия могут привести к удалению параметров или связей, что может сказаться на производительности и результативности модели.

Тем не менее, несмотря на имеющиеся недостатки, данный способ также рассматривается в статье, так как при его использовании даже при минимальных значениях прореживания в совокупности с другими способами оптимизации может дать приемлемые по точности и производительности результаты.

4 Оптимизация вычислений

Оптимизация вычислений также является важным аспектом, она может включать в себя использование более эффективных алгоритмов, оптимизацию вычислительных графов,

распределение вычислений на графический процессор (GPU) или использование специализированного аппаратного ускоритель (например, Tensor Processing Unit) [8]. Особенностью данного чипа является то, что он специально был спроектирован для работы с моделями и обработкой многомерных данных. Упрощенная схема тензорного процессора от компании «Nvidia» приведена на рисунке 3.

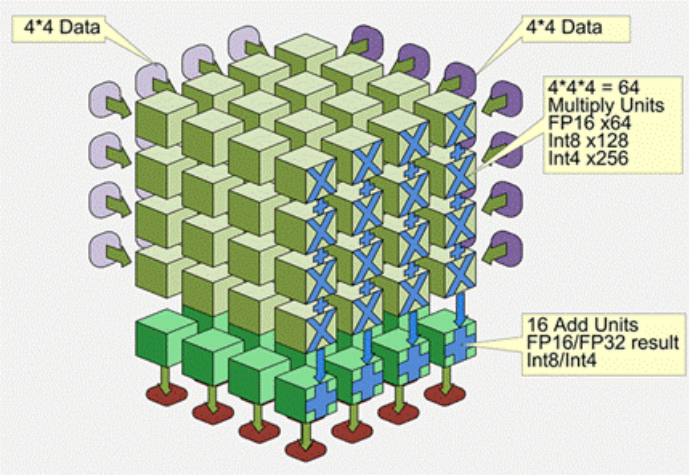

Рисунок 3 – Тензор (TPU) от компании «Nvidia»

Оптимизация вычислений может привести к значительному ускорению работы алгоритмов машинного обучения. Использование более эффективных алгоритмов, оптимизация вычислительных графов и распределение вычислений на специализированные аппаратные ускорители, могут значительно улучшить производительность. Однако эти методы требуют хорошего понимания алгоритмов и вычислительных моделей, а также опыта в их реализации и оптимизации.

5 Выбор фреймворков и инструментов машинного обучения

Кроме того, существуют фреймворки и инструменты, специально разработанные для оптимизации производительности машинного обучения на iOS. Некоторые из них включают библиотеки «CoreML» [9], «Metal Performance Shaders», а также фреймворк «Metal API» и другие.

На рисунке 4 приведена схема работы фреймворка «CoreML», суть которого заключается в преобразовании из обычной модели для стационарных вычислительных устройств в специальный оптимизированный формат под мобильные устройства, который обрабатывается непосредственно самой библиотекой и поставляется в разрабатываемое мобильное приложение.

Рисунок 4 – Схема работы «CoreML»

Фреймворк «Metal Performance Shaders» содержит коллекцию высокооптимизированных вычислительных и графических шейдеров, разработанных для легкой и эффективной интеграции в мобильное приложение. Эти параллельные по данным примитивы специально настроены для максимального использования уникальных аппаратных

особенностей каждого семейства графических процессоров (GPU) с целью обеспечения оптимальной производительности. Приложения, использующие фреймворк «Metal Performance Shaders», достигают отличной производительности без необходимости создания и поддержки ручных шейдеров для каждого семейства GPU. «Metal Performance Shaders» могут использоваться вместе с другими существующими ресурсами вашего приложения (такими как объекты MTLCommandBuffer, MTLTexture и MTLBuffer) и шейдерами [9]. Фреймворк поддерживает следующий функционал:

- применение высокопроизводительных фильтров к изображениям и извлечение статистических и гистограммных данных из них,

- реализация и запуск нейронных сетей для обучения и вывода машинного обучения,

- решение систем уравнений, факторизация матриц и умножение матриц и векторов [10][11],

- ускорение трассировки лучей с помощью высокопроизводительного тестирования пересечений лучей и геометрии.

В свою очередь библиотека «Metal» —низкоуровневый, малозатратный программный интерфейс для аппаратного ускорения 3D-графики и вычислений, разработанный Apple и дебютировавший в iOS 8. «Metal» объединяет функции, схожие с OpenGL и OpenCL, в одном комплекте. Он предназначен для повышения производительности, предоставляя низкоуровневый доступ к аппаратным возможностям графического процессора (GPU) для приложений на iOS, iPadOS, macOS и tvOS. Его можно сравнить с низкоуровневыми API на других платформах, такими как Vulkan и DirectX 12. Metal является объектно-ориентированным, что позволяет использовать его с такими языками программирования, как Swift, Objective-C или C++17. Согласно промо-материалам Apple: MSL [Metal Shading Language] представляет из себя единый язык, который позволяет более тесно интегрировать графические и вычислительные программы [12].

Существуют также аналоги этих библиотек и для устройств на базе операционной системы Android, однако они не будут рассмотрены ввиду того, что в рамках статьи сделан упор именно на устройства от компании Apple и их чипы с префиксом «A».

Эти инструменты предоставляют оптимизированные функции и API, которые позволяют эффективно использовать аппаратные возможности устройств. Однако использование этих фреймворков требует дополнительных усилий для интеграции существующих моделей и алгоритмов, а также изучения их особенностей и возможностей.

6 Проведение экспериментов по совмещению алгоритмов

Для проведения экспериментов по оптимизации алгоритмов машинного обучения на iOS рассмторим методологию, основанную на систематическом исследовании различных параметров и настроек алгоритмов. Ключевым этапом экспериментов было определение оптимальных значений параметров, таких как «learning rate», «batch size» и количество эпох обучения. Данные параметры были выбраны в связи с тем, что они наибольшим образом влияют на скорость и качество обучения алгоритмов. Так же было уделено особое внимание выбору оптимальной структуры сети и алгоритма оптимизации, адаптированных специально под платформу iOS, что позволило добиться значительного улучшения производительности алгоритмов машинного обучения именно на устройствах iOS.

Для получения наиболее точного результата с наиболее производительными алгоритмами, были проведены следующие эксперименты:

- Попарно рассмотрены каждый из перечисленных способов оптимизации нейронных сетей [12] – квантизация [13], сжатие, применение TPU и используемый фреймворк. Были перепробованы различные комбинации в поиске наилучшей эффективности, часть примерных вариантов комбинаций приведена в таблице 1.

- Ввиду того, что квантизация и сжатие довольно критично снижают точность нейронных сетей [14][15] – был проведен отдельный замер без их использования – только применение чипа TPU и две отдельные комбинации с различными фреймворками = CoreML и Metal

Таблица 1 – Примерные варианты совмещения способов оптимизации

Квантизация	Сжатие	TPU	Фреймворк
Отсутствует	0%	Используется	CoreML
Малая	25%		
Средняя	50%	Не используется	Metal
Сильная	75%		

Полученный результат подтвердил гипотезу о том, что применение алгоритмов сжатия и квантизации радикальным образом снижает точность первоначальной нейронной сети – точность алгоритма машинного обучения упала в несколько раз, хотя и скорость работы выросла на порядок.

В то же время, второй эксперимент дал хорошие результаты – совмещение применения чипа TPU и фреймворков CoreML и Metal дали только рост производительности, без снижения точности – за одной лишь особенностью: каждый из фреймворков необходимо использовать только для решения подходящих под них задач, а именно применение алгоритмов машинного обучения с CoreML и обработка 2D/3D изображений и моделей с помощью фреймворка Metal.

В ходе проведения экспериментов так же выявилось еще одно возможное направление для оптимизации – разбиение процесса обработки на 2 составные части, для CPU и GPU с подходящими друг другу классами эквивалентности.

Заключение

В статье были представлены основные способы оптимизации алгоритмов машинного обучения, однако для достижения наибольшего результата необходимо использовать синтез нескольких из описанных подходов.

После проведения экспериментов с описанными алгоритмами оптимизации, была предпринята попытка объединения всех имеющихся способов оптимизация для достижения наилучшего показателя и решения описанной ранее проблемы – недостаточной эффективности отдельно взятых способов оптимизаций [16][17].

Проведенные исследования и необходимые эксперименты выявили, что комбинация сжатия и квантизации радикальным образом уменьшает точность первоначальной нейронной сети. Таким образом, для достижения оптимизации с допустимыми потерями точности рекомендуется использовать только один из способов оптимизации самого алгоритма. Эмпирическим путем было подтверждена рекомендация разработчиков

инструментария, о том, что необходимо комбинировать возможности специального чипа с одним из фреймворков [18].

Еще одним результатом экспериментов стало выявление нового направление для оптимизации - разбиение процесса обработки входных данных на классы эквивалентности таким образом, чтобы обработка происходила параллельно не только с использованием мощностей GPU и TPU, а также CPU. Примерная условная схема разбиения приведена на рисунке 5. Несмотря на то, что центральный процессор не предназначен для выполнения такого рода операций, при правильном разбиении это дало рост скорости выполнения.

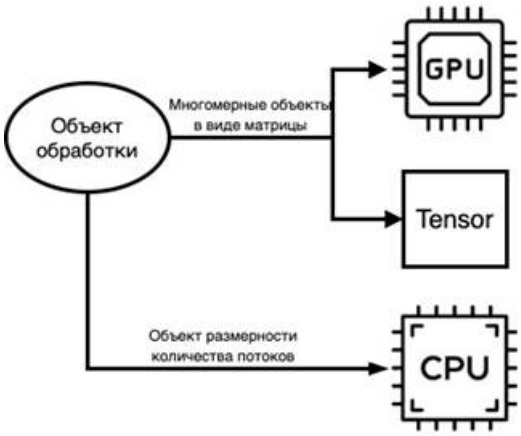

Рисунок 5 – Разбиение объекта обработки по классам эквивалентности

В качестве дальнейших направлений работы планируется исследовать применение предложенного метода к мобильный устройствам под операционной системой Android [19], а также реализовать на практике синтез нескольких алгоритмов оптимизации моделей машинного обучения.

Для практической реализации подобного решения будет проведено отдельное исследование и сравнение производительности различных алгоритмов машинного обучения на мобильных устройствах iOS в рамках диссертации и научной работы. Основная цель же этой статьи – определить эффективность и сравнить производительность различных алгоритмов, чтобы выявить наиболее подходящие для использования на iOS устройствах. Для этого были использованы стандартные наборы данных, такие как «MNIST» для распознавания рукописных цифр и «ImageNet»[20] для классификации изображений, чтобы иметь возможность сравнивать результаты с другими исследованиями. В ходе экспериментов будут учитывались различные факторы, которые могут влиять на производительность алгоритмов, такие как размер набора данных, сложность модели и выбранные параметры. Были проведены эксперименты с различными настройками фреймворков и библиотек, используемых с алгоритмами машинного обучения на мобильных устройствах, и параметрами для оценки их влияния на производительность.

Библиография

1. Чжан Я., Лю Я., Чэнь Т., Генг У. "Мобильное глубокое обучение для интеллектуальных мобильных приложений: Обзор." // IEEE Access, 8,103,586-103,607.
2. Документация Apple Developer [Электронный ресурс] // "Core ML - Оптимизация производительности на устройствах." // URL: https://developer.apple.com/documentation/coreml/optimizing_for_on-device_performance. Дата обращения: [29.06.2023].

3. Растегари М., Ордонес В., Редмон Дж., Фархади А. "XNOR-Net: Классификация изображений ImageNet с использованием бинарных сверточных нейронных сетей." // Сборник материалов Европейской конференции по компьютерному зрению (ECCV) (стр. 525-542).
4. Сихотан Х.., Марк А., Риандари Ф. и Ренделл Л. "Эффективные алгоритмы оптимизации для различных задач машинного обучения, включая классификацию, регрессию и кластеризацию. " // IEEE Access, 1. 14-24. 10.35335/idea.v1i1.3.
5. Сандлер М., Ховард А., Чжу М., Жмогинов А., Чен Л. Ц. "MobileNetV2: Инвертированные остаточные блоки и линейные узкие места." // Сборник материалов конференции по компьютерному зрению и обработке изображений // IEEE (стр. 4510-4520).
6. Ховард А. Г., Чжу М., Чен Б., Калениченко Д., Ванг В., Веанд Т. И др. "MobileNets: Эффективные сверточные нейронные сети для мобильных приложений компьютерного зрения." // IEEE 1704.04861.
7. Хан С., Мао Х., Дэлли У. Дж. "Глубокая компрессия: Сжатие глубоких нейронных сетей с помощью отсечения, квантования обучения и кодирования Хаффмана." // IEEE 1510.00149.
8. Документация Google TensorFlow Lite. [Электронный ресурс]. «TensorFlow» // URL: https://www.tensorflow.org/lite. Дата обращения: [29.06.2023].
9. Таккар М. "Начало машинного обучения в iOS: CoreML Framework." // IEEE Access, 10.1007/978-1-4842-4297-1. // ISBN: 978-1-4842-4296-4
10. Минитаева, А. М. Принятие решений в условиях интервального задания предпочтений лиц, принимающих решений / А. М. Минитаева // Материалы конференции «Информационные технологии в управлении» (ИТУ-2022) : 15-я МУЛЬТИКОНФЕРЕНЦИЯ ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ, Санкт-Петербург, 04 06 октября 2022 года. – Санкт-Петербург: Концерн; Центральный научно-исследовательский институт ;Электроприбор;, 2022. – С. 197-200. – EDN RNGSXI.
11. Минитаева, А. М. Многомодельный подход к прогнозированию нелинейных нестационарных процессов в задачах оптимального управления / А. М. Минитаева // Необратимые процессы в природе и технике : Труды Двенадцатой Всероссийской конференции. В 2-х томах, Москва, 31 января – 03 2023 года. – Москва: Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), 2023. – С. 438-447. – EDN FBVVXS.
12. Кочнев А., «Концептуальные основы практического использования нейронных сетей: проблемы и перспективы» // «Общество и инновации» // DOI: 10.47689/2181-1415-vol4-iss1-pp1-10
13. Курбариа М., Бенгио Й., Дэвид Ж. П. "BinaryNet: Обучение глубоких нейронных сетей с ограничениями на веса и активации +1 или -1." // IEEE: 1602.02830.
14. Ли Г., Вэй Гао, Вуэн Г., «Техники квантизации» // DOI: 10.1007/978-981-97-1957-0_5
15. Самсиана С., Сьямсул А. «Алгоритмы машинного обучения с использованием метода векторного квантования обучения» // DOI: 10.1051/e3sconf/202450003010
16. Адереми А. Атайеро, Сэмюэл Аджани «Обзор машинного обучения на встроенных и мобильных устройствах: оптимизация и приложения» // DOI: DOI:10.3390/s21134412
17. Сандлер М., Ховард А., ЛеКун Й. "Mobilenetv3: Высокоэффективная масштабируемая модель мобильного компьютерного зрения." // Сборник материалов конференции по компьютерному зрению и обработке изображений // IEEE/CVF (стр. 13840-13848).
18. Чен Б., Данда Р. Юан Ч. «На пути к краже глубоких нейронных сетей на мобильных устройствах» // Безопасность и конфиденциальность в сетях связи (стр. 495-508) // DOI:10.1007/978-3-030-90022-9_27
19. Джармуни Ф., Фавзи А. «Запуск нейронных сетей в Android» // Университет Оттавы.

Введение в глубокое обучение и нейронные сети с Python (cnh/ 247-280) //
DOI:10.1016/B978-0-323-90933-4.00001-2
20. Быков К., Мюллер К. «Опасности изображений с водяными знаками в ImageNet» //
Искусственный интеллект. Международные семинары ECAI 2023 (стр. 426–434) //
DOI:10.1007/978-3-031-50396-2_24

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Статья посвящена исследованию и оптимизации алгоритмов машинного обучения для распознавания изображений на мобильных устройствах под управлением iOS. В работе рассматриваются различные методы оптимизации, такие как квантизация и сжатие моделей, а также использование специализированных фреймворков и инструментов для повышения производительности.
Методология исследования включает анализ существующих решений, проведение экспериментов для проверки эффективности различных методов оптимизации, а также сравнительный анализ производительности алгоритмов. Авторы использовали конкретные инструменты и фреймворки, такие как CoreML и Metal Performance Shaders, для тестирования и оптимизации моделей машинного обучения.
Актуальность работы обусловлена широким распространением мобильных устройств и возрастающим спросом на приложения, использующие машинное обучение. Ограниченные вычислительные ресурсы, энергопотребление и необходимость обеспечения высокой производительности на мобильных устройствах делают данную тему весьма значимой и востребованной.
Научная новизна статьи заключается в предложении комплексного подхода к оптимизации производительности алгоритмов машинного обучения на мобильных устройствах iOS. В статье представлены новые комбинации методов оптимизации, такие как совместное использование квантизации и сжатия моделей, что позволяет достигать более высоких показателей производительности и эффективности.
Стиль изложения статьи научный, текст хорошо структурирован. Статья включает введение, обзор существующих решений, описание методов оптимизации, результаты экспериментов и заключение. Каждая часть логически связана с предыдущей, что облегчает восприятие материала. Содержание статьи соответствует заявленной теме и охватывает все ключевые аспекты исследования.
Библиография содержит актуальные и релевантные источники, включая научные статьи и документацию по используемым фреймворкам и методам оптимизации. Однако рекомендуется добавить больше ссылок на современные исследования и публикации, связанные с мобильными приложениями и машинным обучением.
Авторы подробно рассматривают недостатки и ограничения предложенных методов, что показывает их объективность и стремление к всестороннему анализу проблемы. В статье приведены сравнения с аналогичными решениями, что укрепляет аргументацию и научную значимость работы.
Выводы статьи логичны и обоснованы. Авторы суммируют результаты экспериментов и предлагают направления для дальнейших исследований. Практическая значимость работы заключается в возможности применения предложенных методов оптимизации в реальных мобильных приложениях, что будет интересно разработчикам и исследователям в области машинного обучения и мобильных технологий.
Рекомендации по доработке:

1. Уточнить методологию проведения экспериментов, добавить больше подробностей о параметрах и настройках используемых алгоритмов.
2. Увеличить количество современных источников в библиографии для более полного отражения текущего состояния исследования.
3. Расширить раздел о практическом применении предложенных методов, включив больше примеров и кейсов.
4. Включить обсуждение возможных ограничений и потенциальных рисков при использовании предложенных методов оптимизации в реальных условиях.

Статья представляет собой значимый вклад в область оптимизации алгоритмов машинного обучения на мобильных устройствах. Она обладает научной новизной, актуальностью и практической значимостью. При выполнении вышеуказанных рекомендаций работа может быть рекомендована к публикации.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Предмет исследования. С учётом сформированного заголовка статья должна быть посвящена вопросам оптимизации производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS.

Методология исследования базируется на анализе и синтезе данных. Ценно, что автор применяет графический инструментарий представления полученных результатов. При этом обращает на себя внимание отсутствие указанных источников под таблицами и рисунками. Особое внимание автор уделяет проведению эксперимента, который подтвердил авторскую гипотезу.

Актуальность исследования вопросов, сопряжённых с оптимизацией производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS, не вызывает сомнения, т.к. вопросы цифровизации социально-экономических процессов обеспечивает ускорение их осуществления, что в том числе отражается и на экономии финансовых ресурсов. Потенциальную читательскую аудиторию интересуют возможности применения полученных результатах в решении задачи по обеспечению технологического суверенитета Российской Федерации.

Научная новизна в представленном на рецензирование материале присутствует. Например, связана с обоснованием тезиса о том, что «применение алгоритмов сжатия и квантизации радикальным образом снижает точность первоначальной нейронной сети – точность алгоритма машинного обучения упала в несколько раз, хотя и скорость работы выросла на порядок». Также было бы выигрышно в тексте статьи указать потенциальную читательскую аудиторию и конкретные направления использования полученных результатов.

Стиль, структура, содержание. Стиль изложения научный. Структура статьи автором выстроена. Рекомендуется также добавить блок «Обсуждение полученных результатов», а также часть заключения трансформировать в раздел «Дальнейшие направления научных исследований». Ознакомление с содержанием показало логичное изложение

материала в рамках заявленных структурных элементов.

Библиография. Автором сформирован библиографический список из 20 наименований. Ценно, что в нём есть как отечественные, так и зарубежные авторы. Также было бы интересно изучить конкретные статистические данные, описывающие практику применения алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств за последние годы. Это позволило бы автору дополнительно обосновать актуальность исследования с применением конкретного числового обоснования.

Апелляция к оппонентам. Несмотря на сформированный список научных публикаций, какой-либо научной дискуссии в тексте рецензируемой научной статьи не обнаружено. При доработке статьи автору рекомендуется уделить внимание устранению данного замечания. Это позволит автору конкретно показать наличие прироста научного знания, который однозначно автором сделан, но очень многое из текущей редакции не воспринимается настолько выигрышно, как могло бы быть представлено.

Выводы, интерес читательской аудитории. С учётом всего вышеизложенного заключаем о том, что статья подготовлена на высоком уровне, обладает научной новизной и практической значимостью. Доработка статьи по указанным в тексте замечаниям позволит ещё больше расширить потенциальную читательскую аудиторию.