УДК 004

ИССЛЕДОВАНИЕ СПОСОБОВ ОПТИМИЗАЦИИ СКОРОСТИ РАБОТЫ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ НА КЛИЕНТСКОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЙ

Ермолаева А.А. (ИТМО)
Научный руководитель – преподаватель, кандидат физико-математических наук Жуков Н.Н. (ИТМО)

Введение. Развитие машинного обучения привело к его повсеместному использованию в различных сферах жизни. Однако для достижения лучших результатов системе необходимо совершить большое количество тяжеловесных вычислений, так как с ростом количества параметров и данных растут требования к производительности, необходимой для моделирования, обучения и развертывания больших моделей. Данная проблема становится актуальна и для веб-приложений, сайтов, в которых используется глубокое обучение для обработки изображений, распознавании голоса, веб-поиске, системах рекомендаций [1].

Основная часть. Задачи машинного обучения требуют использования большого количества ресурсов по памяти и производительности. Использование предварительно обученных моделей и их дообучение на новых данных может значительно сократить время обучения. Однако существуют и другие варианты оптимизации. Модели машинного обучения на стороне клиента не требуют отправки запросов на сервер, по данной причине они должны иметь меньшие задержки. Однако на практике из-за аппаратных ограничений задержки могут получиться больше. Для сокращения времени ожидания результата работы модели существуют аппаратные способы оптимизации, такие как WebGL и WebAssembly.

Изначально WebGL создавался как инструмент для отрисовки 3D графики, но затем API стали использовать для выполнения операций не связанных с рендерингом [2]. Упаковка и распаковка тензорных данных в текстуру, отсутствие общего доступа к памяти между вычислениями привело к повторению одной и той же работы, а следовательно, к потере производительности. В WebGPU появляется такое понятие, как вычислительные шейдеры. Вычислительные шейдеры не привязаны к структуре операций рендеринга, а также могут обмениваться данными и результатами вычисления в группах шейдеров, что повышает эффективность целой системы.

Для оптимизации вычислений на клиентской стороне приложений существует технология Web-Worker [3]. Web-Worker — это средство для запуска сценариев в фоновых потоках, позволяющее выполнять сложные вычисления в фоновых потоках, без блокировки пользовательского интерфейса. Благодаря фоновым потокам выполнение инструкций не мешает приложению оперативно реагировать на действия пользователя в приложении.

Сравнение показателей производительности и применение оптимизаций производилось на предварительно разработанном приложении по считыванию эмоций человека с помощью библиотеки Human. Обработка каждого кадра происходила посредством передачи в Web-Worker буфера изображения и его размеров, где выполнялся процесс обнаружение лица и распознавания эмоций, а полученный результат распознавания отправлялся в основной поток приложения.

Выводы. В ходе эксперимента было выявлено, что использование технологий Web-Worker и WebGPU позволяет сократить время ожидания получения результата работы модели машинного обучения. При использовании данных оптимизаций в приложении по распознаванию эмоций получилось добиться сокращения времени ожидания результата на 1,5 секунды. Основная проблема с развертыванием моделей машинного обучения на мобильных устройствах и в браузерах связана с аппаратными ограничениями и ограничениями

интеграции, поэтому предложенное решение позволяет минимизировать потери в производительности.

Список использованных источников:

1. Stocco A. How artificial intelligence can improve web development and testing // Proceedings of the 3rd International Companion Conference on Art, Science, and Engineering of Programming. – 2019. – №13. – С. 1-4.

2. WebGPU [Электронный ресурс] URL: https://developer.chrome.com/blog/webgpu-io2023 (дата обращения: 22.01.2024).

3. Web Workers API [Электронный ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API (дата обращения: 22.01.2024).