Научная статья
УДК 004.8
URL: https://trudymai.ru/published.php?ID=189700
EDN: https://www.elibrary.ru/CNNCGF

ОПТИМИЗИРОВАННЫЙ МОДУЛЬ GRU С ГЛОБАЛЬНЫМ КОНТЕКСТОМ В ВИЗУАЛЬНОЙ ОДОМЕТРИИ

П.Г. Корыткин ✉

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В. Ломоносова», г. Москва, Россия

✉korytkinpg@my.msu.ru

Цитирование: Корыткин П.Г. Оптимизированный модуль GRU с глобальным контекстом в визуальной одометрии // Труды МАИ: электрон. журнал. № 150.
URL: https://trudymai.ru/published.php?ID=189700

Аннотация. Применение рекуррентных модулей в нейросетевых архитектурах остаётся необходимым для решения задач, связанных с наборами данных, изменяющихся во времени, для достижения оптимальных результатов по точности. При этом, в случае применения таких модулей в задачах, связанных с компьютерным зрением на беспилотных летательных аппаратах и робототехнических комплексах, необходим больший объём памяти, чем имеющийся из-за того, что изображения требуют большего числа весов и процессорных ресурсов, поэтому остаётся актуальным оптимизация таких модулей для применения в задачах компьютерного зрения реального времени. В работе рассматривается оптимизация свёрточного варианта модуля Gated Recurrent Unit, с сохранением глобального контекста и жёсткой сигмоидой, для экономии числа параметров и вычислительных ресурсов, применительно к задаче визуальной одометрии и архитектуре DROID-SLAM. Дополнительно проверен выигрыш по скорости выполнения при применении жёсткой сигмоиды относительно обычной сигмоиды на процессорах: Ryzen 5500U, Ryzen 5950X и на

платформе Google Colab, а также на видеоадаптерах Nvidia RTX 5070 Ti и Nvidia Tesla T4. Также выполнен анализ потребления памяти для сравнения оригинального и модифицированного модуля в различных модификациях. Полученная архитектура с модифицированным рекуррентным модулем Gated Recurrent Unit обучена на наборе данных TartanAir состоящего из синтетических изображений с камеры малого беспилотного летательного аппарата и робототехнических комплексов на 250000 шагах. В силу того, что было снижено потребление памяти, удалось использовать на обучении видеоускоритель с 16 Гб памяти вместо 24 Гб. Платформа для обучения и тестирования состояла из Nvidia RTX 5070 Ti и Ryzen 5950X с 64 Гб RAM 2667 МТ/с. Проверка выполнена на наборах EuRoC и TUM содержащих реальные изображения, захваченные с камер малых беспилотных летательных аппаратов. В результате, выполнения работы по оптимизации, удалось достичь снижения времени выполнения до 20% на Nvidia RTX 5070 Ti. Общее число параметров у архитектуры DROID-SLAM снизилось на 27% от исходных 4002565 параметров к 2937093. Число параметров в полученном модуле Gated Recurrent Unit снизилось на ~66% по сравнению с оригинальным GRU в DROID-SLAM, то есть с 1614720 до 549248.

Ключевые слова: визуальная одометрия; нейронные сети; GRU; оптимизация числа параметров; DROID-SLAM.

OPTIMIZED GRU MODULE WITH GLOBAL CONTEXT IN VISUAL ODOMETRY

P.G. Korytkin

Federal State Budget Educational Institution of Higher Education M.V. Lomonosov Moscow State University, Moscow, Russia

korytkinpg@my.msu.ru

Citation: Korytkin P.G. Optimized GRU module with global context in visual odometry // Trudy MAI. 2026. No. 150. (In Russ.). URL: https://trudymai.ru/published.php?ID=189700

Abstract. The application of recurrent modules in neural network architectures remains essential for addressing tasks involving time-varying datasets, as they are key to achieving optimal accuracy. However, when such modules are applied to computer vision

tasks for unmanned aerial vehicles (UAVs) and robotic systems, the memory requirements often exceed available resources because image data necessitates a larger number of weights and greater computational resources. Consequently, optimizing these modules for real-time computer vision applications remains a critical challenge. This work explores the optimization of a convolutional variant of the Gated Recurrent Unit (GRU) - incorporating global context and a hard sigmoid activation function - to reduce both the number of parameters and computational resource consumption, specifically within the context of visual odometry and the DROID-SLAM architecture. Furthermore, we evaluate the performance gains (in terms of execution speed) achieved by using a hard sigmoid instead of a sigmoid across various hardware platforms, including Ryzen 5500U and Ryzen 5950X processors, Google Colab, and GPUs such as the Nvidia RTX 5070 Ti and Nvidia Tesla T4. We also analyze memory consumption, comparing the original module against the modified version. The resulting architecture, featuring the optimized GRU, was trained for 250000 steps on the TartanAir dataset, which consists of synthetic imagery simulating data from cameras mounted on small UAVs and robotic systems. Due to the reduction in memory usage, it became possible to perform training on a GPU with 16 GB of VRAM, rather than the 24 GB otherwise required. The platform used for training and testing consisted of an Nvidia RTX 5070 Ti and a Ryzen 5950X, paired with 64 GB of RAM running at 2667 MT/s. Evaluation was conducted using the EuRoC and TUM datasets, which contain real-world imagery captured by cameras mounted on small unmanned aerial vehicles (UAVs). As a result of the optimization efforts, execution time on the Nvidia RTX 5070 Ti was reduced by up to 20%. The total number of parameters in the DROID-SLAM architecture decreased by 27% - from an initial 4002565 to 2937093. The number of parameters in the resulting Gated Recurrent Unit (GRU) module was reduced by approximately 66% compared to the original GRU in DROID-SLAM, dropping from 1614720 to 549248.

Keywords: visual odometry; neural networks; GRU; parameter number optimization; DROID-SLAM.

Введение

Применение нейронных сетей в задачах с реальным временем сопряжено с необходимостью оптимизаций архитектуры для повышения скорости работы и снижения потребляемых ресурсов, в том числе из-за того, что на беспилотных летательных аппаратах имеются ограничения по энергетике [1]. Проблема энергетики также имеет место в мобильных робототехнических комплексах и космических аппаратах. При этом, в настоящее время на рынке наблюдается дефицит чипов памяти и другого критического оборудования, в связи с активным строительством большого числа центров обработки данных [2, 3], приведшее к удорожанию вычислительных устройств, имеющих в своём составе чипы памяти. В частности отмечается, что за 2026 год дата-центры потребили до 70% чипов памяти [4]. Поэтому актуальным является направление развития нейросетевых технологий, связанное со снижением потребления оперативной и видео памяти [5] на этапах обучения и выполнения, для экономии денежных средств необходимых для закупки оборудования. При этом известно, что на этапе обучения потребление памяти, в большинстве случаев занимает 4x относительно того, что будет на выполнении. Причиной является необходимость хранения в памяти веса модели, веса градиентов и веса всех матриц, по которым будет расчёт градиентов, а также матриц, используемых оптимизатором Adam.

При этом остаётся актуальным применение нейронных сетей в таких задачах, как визуальная одометрия.

Визуальная одометрия (ВО) – вычисление угловых и линейных координат на основе последовательности изображений.

Задачу визуальной одометрии возможно рассматривать, как задачу, связанную с прогнозированием перемещения на основе временного ряда для данных, составляющих изображения с камеры. Поэтому, из-за близости к задаче прогнозирования временных рядов в существующих архитектурах, встречаются рекуррентные модули по типу Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU). Данные рекуррентные модули представляют собой специальную архитектуру, содержащую ячейки памяти, в которых связи между элементами образуют последовательность. Модуль GRU является упрощением LSTM в котором

несколько вентилей объединены в один, что позволяет быстрее проводить обучение, снизить вычислительную сложность, снизить число параметров.

Например, в существующей архитектуре DeepVO [6] встречается LSTM и в статье авторов [7], а в DROID-SLAM [8] - GRU с использованием свёрток. Также GRU встречается в комбинации с фильтром Калмана, например в статье авторов [9]. В статье [11] продолжено развитие DROID-SLAM, но также задействуется модуль GRU. При этом, в статьях [12, 13] используется модифицированный свёрточный модуль GRU, с добавлением модуля на основе сетей называемых Kolmogorov-Arnold Networks (KAN) [14], как смещения. Такие модули позволяют захватить некоторый набор информации из последовательности данных и сохранить в матрицу памяти. Хранение темпоральной информации о сцене является необходимой составляющей для реализации устойчивой визуальной одометрии во времени. Развитие рекуррентных модулей, работающих с временными последовательностями, является актуальной задачей [15].

При этом, важной проблемой является то, что такие модули при работе с изображениями более требовательны к объёму необходимой оперативной памяти и вычислительным ресурсам. Помимо этого, у таких модулей кратно большее число параметров, чем у простых свёрточных слоёв по причине того, что внутри таких модулей содержится комбинация из нескольких других слоёв, каждый из которых содержит обучаемые параметры.

Помимо того, актуальным является применение визуальной одометрии на небольших низколетящих беспилотных летательных аппаратах для корректирования накапливаемой ошибки положения используя отличительные признаки окружения в особенности при условии отсутствия, переотражения или подавления GPS. Поэтому в данном направлении продолжаются исследования, как например в работе авторов [16].

Поэтому направление, связанное с оптимизацией рекуррентных модулей применительно к компьютерному зрению для снижения потребляемых вычислительных ресурсов, остаётся актуальным.

Во всех формулах используется знак $\odot$ для обозначения произведения Адамара, σ для обозначения функции сигмоиды, рассматривается пара моментов времени t и $t-1$. Оператор $*$ обозначает двумерную свёртку.

Модуль GRU записываемый математически с использованием свёрток, имеющих C каналов для изображений высоты H и ширины W определяется с использованием формул 1-7:

$$x_t \in R^{H\times W\times C_m} \tag{1}$$

$$h_t \in R^{H\times W\times C_h} \tag{2}$$

$$\widehat{\mathrm{h}_t} \in R^{H\times W\times C_h} \tag{3}$$

$$z_t \in \sigma(W_{xz} * x_t + W_{hz} * h_{t-1} + b_z) \tag{4}$$

$$r_t \in \sigma(W_{xr} * x_t + W_{hr} * h_{t-1} + b_r) \tag{5}$$

$$\widehat{h_t} \in \tanh(W_x * x_t + W_h * (r_t \odot h_{t-1}) + b_h) \tag{6}$$

$$h_t \in (1 - \mathrm{z_t}) \odot \mathrm{h_{t-1}} + \mathrm{z_t} \odot \widehat{\mathrm{h}_\mathrm{t}} \tag{7}$$

При этом, используя типовые слои $Conv2d$ в PyTorch, свёртка по умолчанию имеет обучаемые матрицы смещения b и весов W внутри слоя свёртки, из-за чего запись упрощается, учитывая, что в каждом слое свёртки своя матрица весов за место одной исходной. В результате получается набор формул 8-12:

$$\mathrm{Conv2d(a)} = \mathrm{W_a} * \mathrm{a} + \mathrm{b_a} \tag{8}$$

$$z_t \in \sigma\big(Conv2d(x) + Conv2d(h_{t-1})\big) \tag{9}$$

$$r_t \in \sigma\big(Conv2d(x) + Conv2d(h_{t-1})\big) \tag{10}$$

$$\widehat{h_t} \in \tanh\big(Conv2d(\mathrm{x_t}) + \mathrm{Conv2d}(\mathrm{r_t} \odot \mathrm{h_{t-1}})\big) \tag{11}$$

$$h_t \in (1 - \mathrm{z_t}) \odot \mathrm{h_{t-1}} + \mathrm{z_t} \odot \widehat{\mathrm{h}_\mathrm{t}} \tag{12}$$

При этом, для оптимизации и уменьшения числа свёрток в DROID-SLAM применяется подход с соединением матриц и применением одной свёртки, чтоб снизить число свёрток с 6 до 3, что представлено в формулах 13-16:

$$z_t \in \sigma\big(Conv2d([x_t, h_{t-1}])\big) \tag{13}$$

$$r_t \in \sigma\big(Conv2d([x_t, h_{t-1}])\big) \tag{14}$$

$$\widehat{h_t} \in \tanh\big(Conv2d([x_t, r_t \odot h_{t-1}])\big) \tag{15}$$

$$h_t \in (1 - \mathrm{z_t}) \odot \mathrm{h_{t-1}} + \mathrm{z_t} \odot \widehat{\mathrm{h}_\mathrm{t}} \tag{16}$$

Но при этом, используется глобальный контекст путём усреднения, который добавляется дополнительно для повышения точности на ~15% и итоговое число свёрток составляет 7. Формулы с использованием глобального контекста 17-22:

$$g_t \in \sigma\big(Conv2d(h_{t-1})\big) \odot h_{t-1} \tag{17}$$

$$\big(\mathrm{glo}_{b,c}\big)_t \in \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} g_{b,c,i,j} \tag{18}$$

$$z_t \in \sigma\big(Conv2d([x_t, h_{t-1}]) + Conv2d(\mathrm{glo})\big) \tag{19}$$

$$r_t \in \sigma\big(Conv2d([x_t, h_{t-1}]) + Conv2d(\mathrm{glo})\big) \tag{20}$$

$$\widehat{h_t} \in \tanh(Conv2d([x_t, r_t \odot h_{t-1}] + Conv2d(\mathrm{glo})) \tag{21}$$

$$h_t \in (1 - \mathrm{z_t}) \odot \mathrm{h_{t-1}} + \mathrm{z_t} \odot \widehat{\mathrm{h}_\mathrm{t}} \tag{22}$$

Ранее, другими авторами, уже предпринимались попытки оптимизировать модуль GRU. Например, в работе [10] авторами был представлен модуль Minimal Gated Unit (MGU) в котором гейт обновления (update gate) и сброса (reset gate) объединены в единый гейт забывания (forget gate).

В данном подходе используются функции гиперболического тангенса и сигмоида, которые являются ресурсозатратными, а также матрицы смещения.

Развитие данного подхода описывалось в работе авторов [17], с дальнейшим применением в области визуальной одометрии к архитектуре DROID-SLAM. Полученная архитектура названа MINI-DROID-SLAM. Изменения авторов позволили прийти к двум уникальным слоям свёрток и отказу от гиперболического тангенса, при сохранении сигмоиды показанное в формулах 23-25:

$$z_t = \sigma\big(\mathrm{Conv2d}([\mathrm{x_t}, \mathrm{h_{t-1}}])\big) \tag{23}$$

$$\widehat{h_\mathrm{t}} = \mathrm{Conv2d}(\mathrm{x_t}) \tag{24}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h_t} \quad (25)$$

При этом, у данного подхода отсутствует извлечение глобального контекста, который позволил авторам [8] увеличить точность.

Постановка задачи

В нашем случае вместо сигмоиды решено использовать функцию жёсткой сигмоиды (Hard sigmoid), которая менее требовательна к вычислительным ресурсам [22] и аппроксимирует обычную сигмоиду.

При этом, решено сохранить глобальный контекст используя одну свёртку для него. В результате число уникальных слоёв свёртки составило три.

Применение жёсткой сигмоиды не является новым, как функции активации, но её сочетание с предложенной структурой с тремя свертками и сохранением глобального контекста в задаче визуальной одометрии рассматривается впервые. Применение линейных аппроксимаций для функций активации и их влияние на итоговую производительность и вычислительную сложность рассматривалось, например в [21].

Полученный оптимизированный модуль GRU определяется следующими формулами:

$$g_t = Hardsigmoid\big(Conv2d(h_{t-1})\big) \odot h_{t-1} \quad (26)$$

$$\big(\mathrm{glo}_{b,c}\big)_t = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} g_{b,c,i,j} \quad (27)$$

$$\widehat{h_t} = (Conv2d([x_t, h_{t-1}] + Conv2d(\mathrm{glo})) \quad (28)$$

$$z_t = \mathrm{Hardsigmoid}\big(\widehat{h_t}\big) \quad (29)$$

$$h_t = (1 - \mathrm{z_t}) \odot \mathrm{h_{t-1}} + \mathrm{z_t} \odot \widehat{\mathrm{h_t}} \quad (30)$$

$$Hardsigmoid(x) = \begin{cases} 0 \text{ для } x \leq -3 \\ 1 \text{ для } x \geq 3 \\ x/6 + 1/2 \text{ остальные} \end{cases} \quad (31)$$

В архитектуре DROID-SLAM проведена замена модуля GRU на полученный оптимизированный модуль GRU. Также в архитектуре снижено число итераций на

этапе обучения для оптимизации позиции, с которыми работает архитектура с 15 до 8 по причине того, что в распоряжении имеется Nvidia RTX 5070 Ti с 16 Гб VRAM и для использования 15 итераций не хватает видеопамяти. Число кадров оставлено, как в оригинале, равное 7. При этом, параметры на экспериментах по числу кадров и числу итераций для архитектуры оставлены, как в оригинальной архитектуре. Поэтому оптимизации связанные со снижением числа итераций на этапе обучения влияют лишь на качество и скорость обучения, не оказывая влияния на итоговое время выполнения.

Эксперименты и сравнение

Для того, чтобы проверить тезис, что жёсткая сигмоида менее требовательна к вычислительным ресурсам проведено сравнение на CPU и CUDA между жёсткой сигмоидой и оригинальной функцией сигмоиды применительно к матрицам размера NxNxN с числом N от 1 до 255 включительно применительно для чисел с плавающей запятой разрядностей 16, 32, 64 обозначаемых соответственно F16, F32, F64. Дополнительно в PyTorch имеется вариант жёсткой сигмоиды с поддержкой Inplace для выполнения применения функции к матрице без создания новой матрицы, для снижения накладных расходов. Для Inplace варианта жёсткой сигмоиды отдельно проведены замеры.

Замеры производились на процессоре и на видеоускорителе различных вариантов перечисленных ниже конфигураций.

Для проверки на CPU:

1. Ryzen 5500U с 32 Гб RAM.
2. Google Colab с 2 ядрами AMD EPYC 7B12 с ~12.7 Гб RAM.
3. Ryzen 5950X с 64 Гб RAM.

Для проверки на CUDA:

1. Google Colab с 2 ядрами AMD EPYC 7B12 с ~12.7 Гб RAM и Nvidia Tesla T4 с 16 Гб VRAM.
2. Ryzen 5950X с 64 Гб RAM и Nvidia RTX 5070 Ti с 16 Гб VRAM.

Методика тестирования заключалась в том, чтобы выполнить 2 операции связанные с применением функции сигмоиды и жёсткой сигмоиды для прогрева,

а затем замер среднего времени выполнения для 100 операций применения функции сигмоиды/жёсткой сигмоиды. Отдельно данный замер повторялся для обычной функции сигмоиды и Inplace варианта функции жёсткой сигмоиды. Важно отметить, что вариант Inplace функции сигмоиды пока отсутствует в PyTorch. Данный набор тестов повторялся для доступных в PyTorch вариантов чисел с плавающей запятой для 16, 32, 64 разрядов. На рисунке 1 и рисунке 2 представлены замеры для процессора Ryzen 5500U с 32 Гб RAM. На рисунках 3 и 4 для процессора Ryzen 5950X и 64 Гб RAM. На рисунках 5 и 6 замеры для 2 ядер процессора в Google Colab. На рисунках 7 и 8 представлены замеры для CUDA на Nvidia RTX 5070 Ti 16 Гб VRAM. На рисунках 9 и 10 представлены замеры для Nvidia Tesla T4 в Google Colab с 16 Гб VRAM.

Отдельно для каждого из замеров построены графики сравнения отношения времени выполнения функции сигмоиды к времени выполнению функции жёсткой сигмоиды.

На рисунке 11 и рисунке 12 представлены соотношения по времени между сигмоидой и жёсткой сигмоидой для процессора Ryzen 5500U с 32 Гб RAM. На рисунках 13 и 14 соотношения по времени между сигмоидой и жёсткой сигмоидой для процессора Ryzen 5950X и 64 Гб RAM. На рисунках 15 и 16 соотношения по времени между сигмоидой и жёсткой сигмоидой для 2 ядер процессора в Google Colab. На рисунках 17 и 18 представлены соотношения по времени между сигмоидой и жёсткой сигмоидой для CUDA на Nvidia RTX 5070 Ti 16 Гб VRAM. На рисунках 19 и 20 представлены соотношения по времени между сигмоидой и жёсткой сигмоидой для Nvidia Tesla T4 в Google Colab с 16 Гб VRAM.

По приведённым данным видно, что:

1. На Ryzen 5500U жёсткая сигмоида в среднем быстрее при F16 в ~3.19 раза, при F32 в 2.24 раза, при F64 в 2.39 раза. Для Inplace варианта жёсткой сигмоиды: F16 ~4.4 раза, F32 ~3.83 раза, F64 ~3.89 раза.

2. На CPU Google Colab жёсткая сигмоида в среднем быстрее при F16 в ~1.73 раза, при F32 в 1.71 раза, при F64 в 1.97 раза. Для Inplace варианта жёсткой сигмоиды: F16 ~2.39 раза, F32 ~3.5 раза, F64 ~4.27 раза.

3. На Nvidia RTX 5070 Ti сигмоида быстрее на F16 в ~1.05 раза, при F32 в 1.05 раза, а при F64 наоборот жёсткая сигмоида в среднем быстрее в 2.44 раза. Для Inplace варианта жёсткой сигмоиды: F16 ~1.44 раза, F32 ~1.56 раза, F64 ~2.72 раза.

4. На Nvidia Tesla T4 сигмоида быстрее на F16 в ~1.02 раза, при F32 приблизительно одинаковый результат, а при F64 жёсткая сигмоида в среднем быстрее в 2.32 раза. Для Inplace варианта жёсткой сигмоиды: F16 ~1.14 раза, F32 ~1.11 раза, F64 ~2.26 раза.

5. Применение Inplace варианта жёсткой сигмоиды на RTX 5070 Ti позволяет экономить по времени при использовании различных вариантов плавающей запятой:

a. F16: в среднем ~30%, в лучшем случае ~70%

b. F32: в среднем ~36%, в лучшем ~73%

c. F64 в среднем ~27%, в лучшем ~45%

6. Inplace вариант жёсткой сигмоиды превосходит по скорости обычную сигмоиду на CUDA и на CPU.

В результате данных замеров установлено, что для CUDA видеоадаптера при использовании варианта Inplace жёсткой сигмоиды, выполняющей вычисления для матрицы без создания новой, будет наблюдаться ускорение. В случае применения жёсткой сигмоиды, создающей новую матрицу для результатов вычислений, возможен вариант отсутствия прироста скорости, либо значительное ускорение в зависимости от размеров матриц. Для процессора в среднем наблюдается значительное ускорение от 2.2 до 4.4 раз.

Дополнительно хотелось бы отметить, что в официальной документации Nvidia CUDA для ряда случаев рекомендуется отказываться от вычислений экспоненты в сторону аппроксимации применительно к ряду задач [23].

По изученным данным применение жёсткой сигмоиды является перспективным, особенно для нейронных сетей, которые после обучения на видеоадаптере могут выполняться на процессоре.

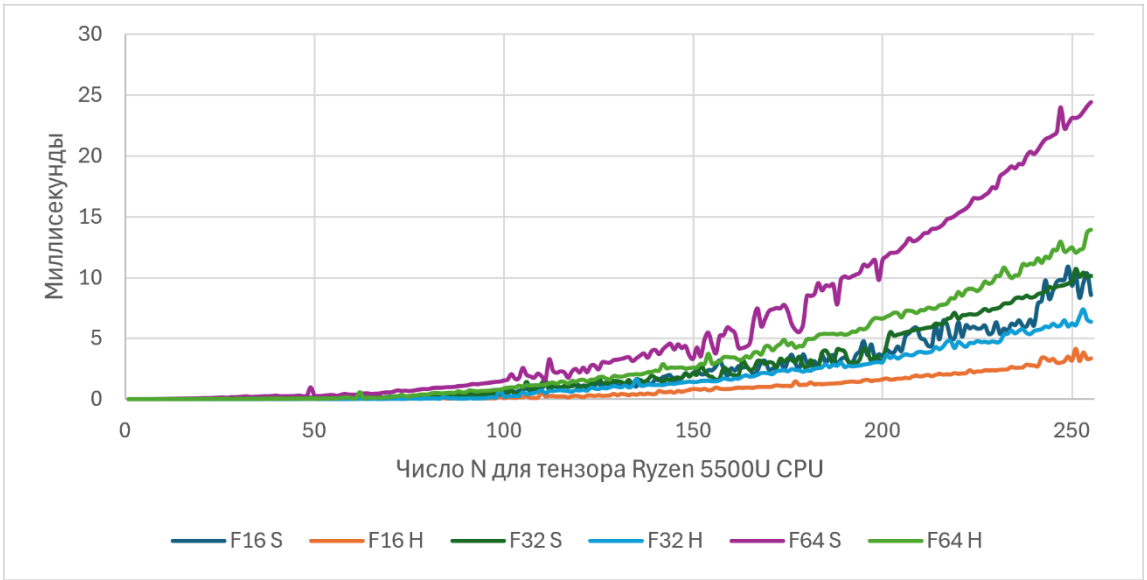


Рисунок 1 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Ryzen 5500U для матриц NxNxN, меньше – лучше

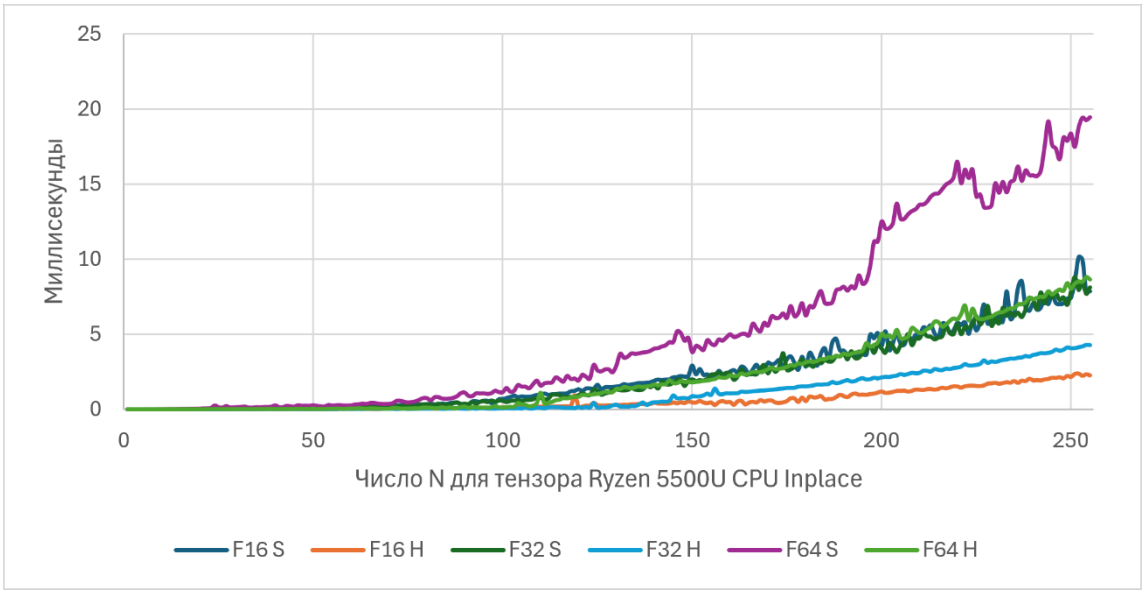


Рисунок 2 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Ryzen 5500U с жёсткой сигмоидой Inplace для матриц NxNxN, меньше – лучше

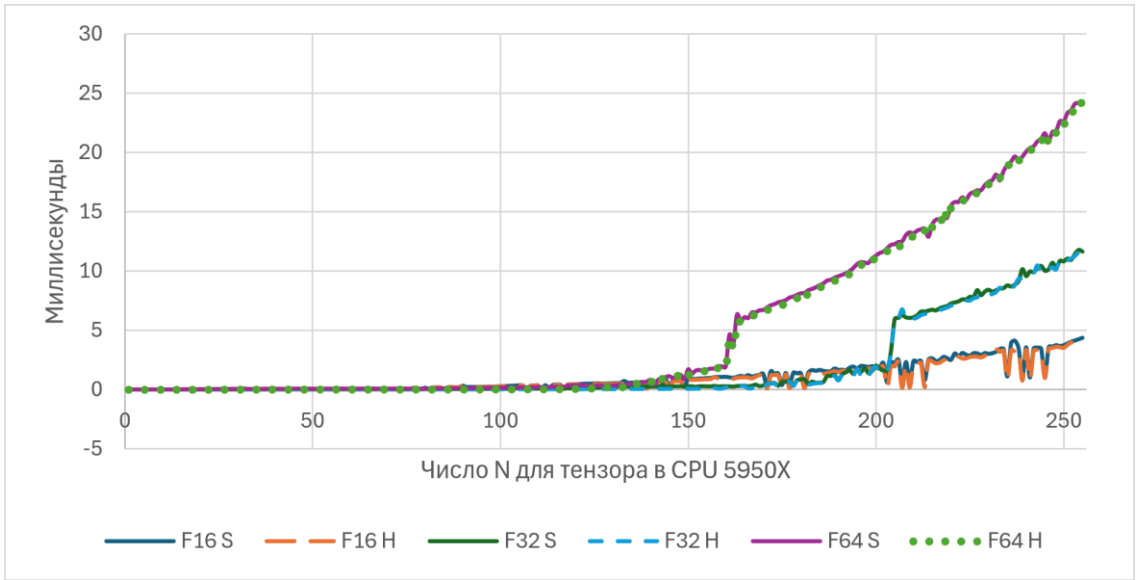


Рисунок 3 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Ryzen 5950X для матриц NxNxN, меньше – лучше

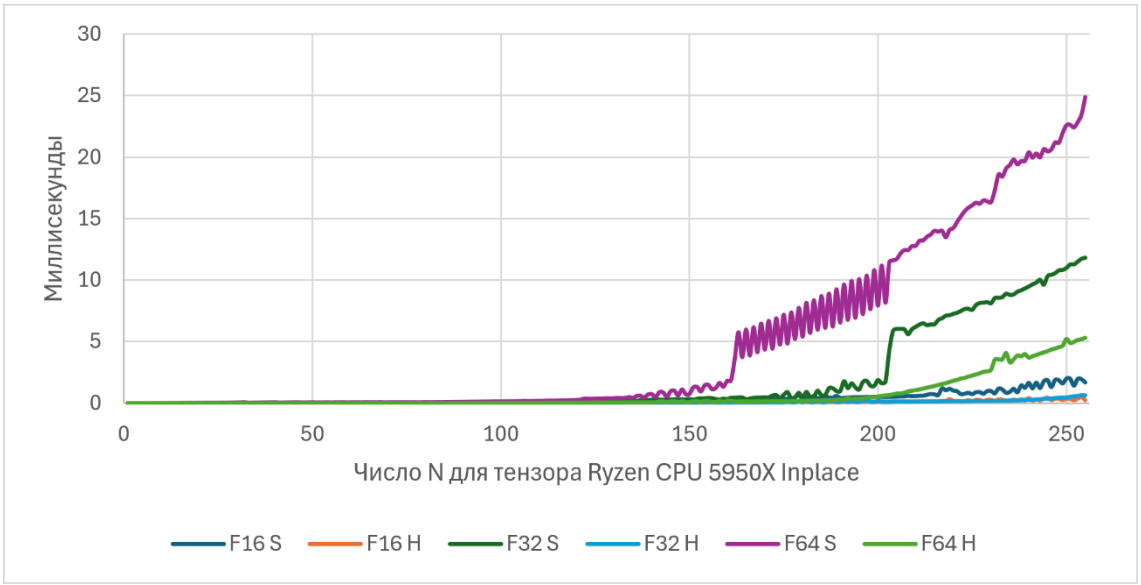


Рисунок 4 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Ryzen 5950X с жёсткой сигмоидой Inplace для матриц NxNxN, меньше – лучше

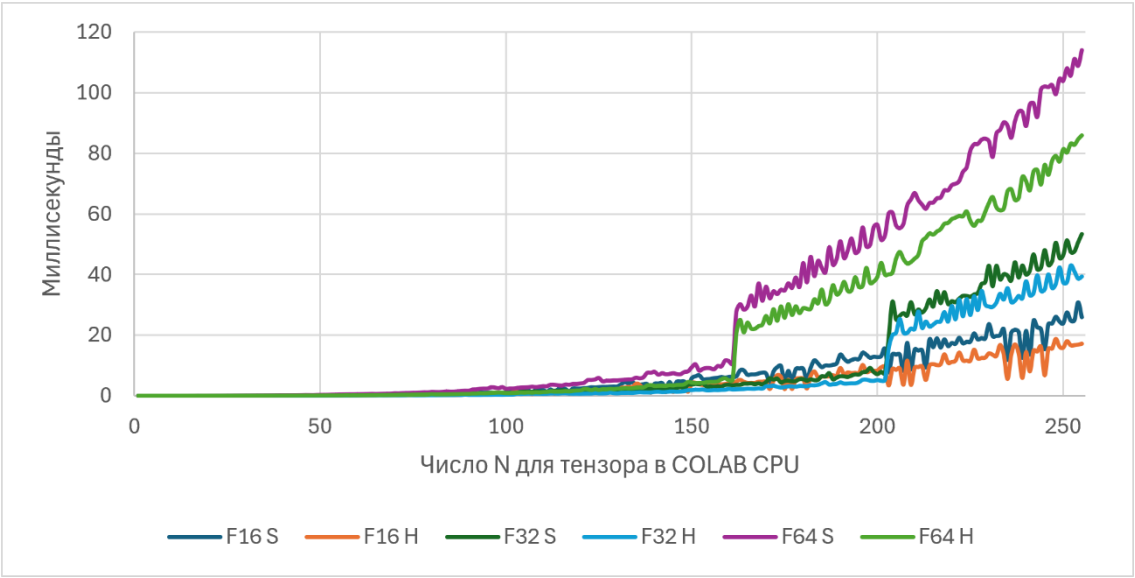


Рисунок 5 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на CPU в Google Colab для матриц NxNxN, меньше – лучше

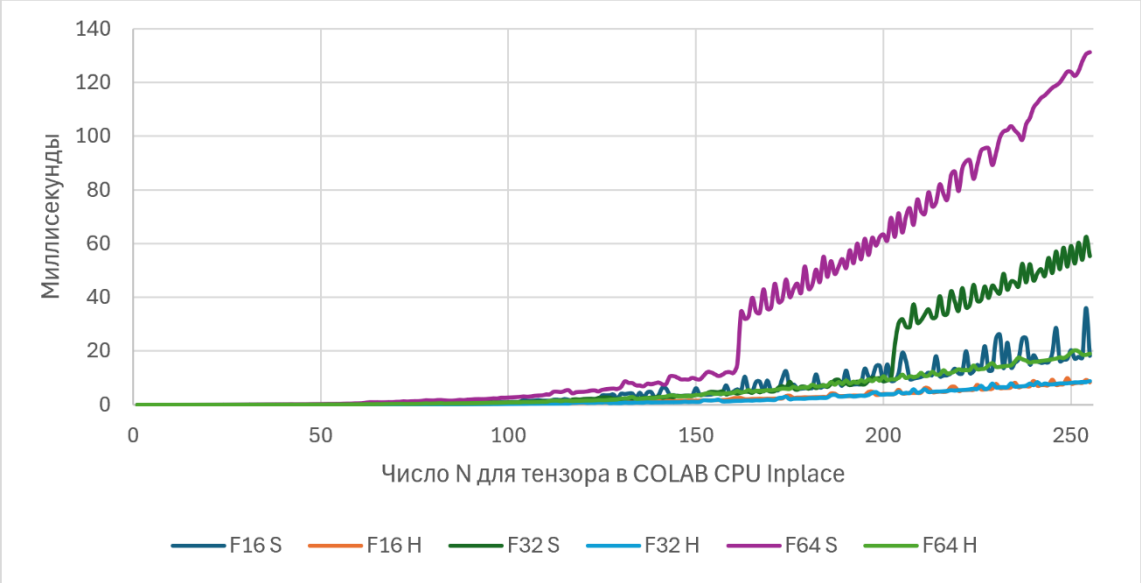


Рисунок 6 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на CPU в Google Colab с жёсткой сигмоидой Inplace для матриц NxNxN, меньше – лучше

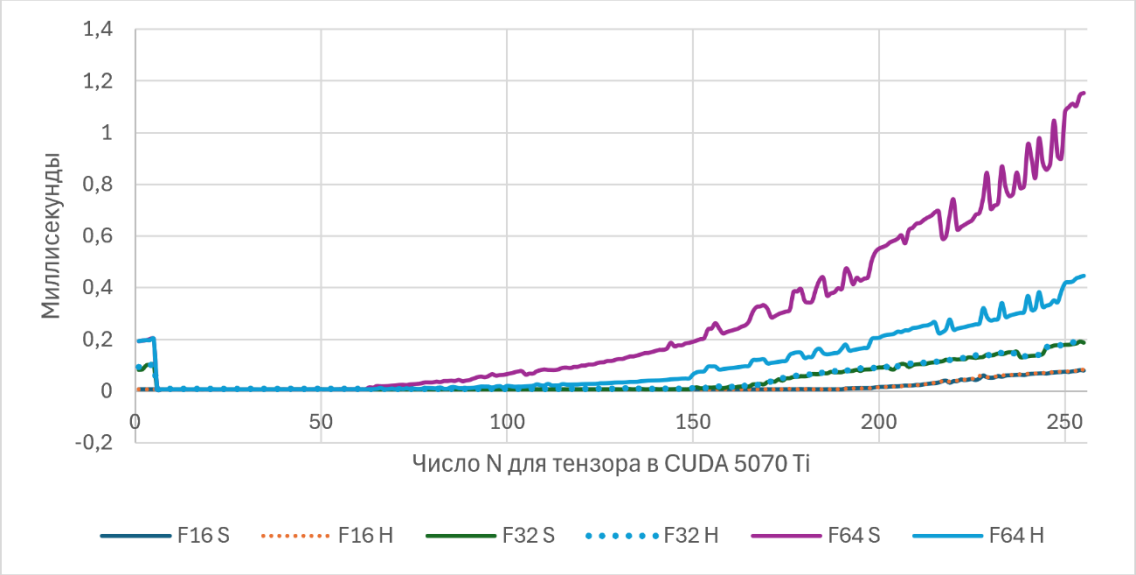


Рисунок 7 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на 5070 Ti для матриц NxNxN, меньше - лучше

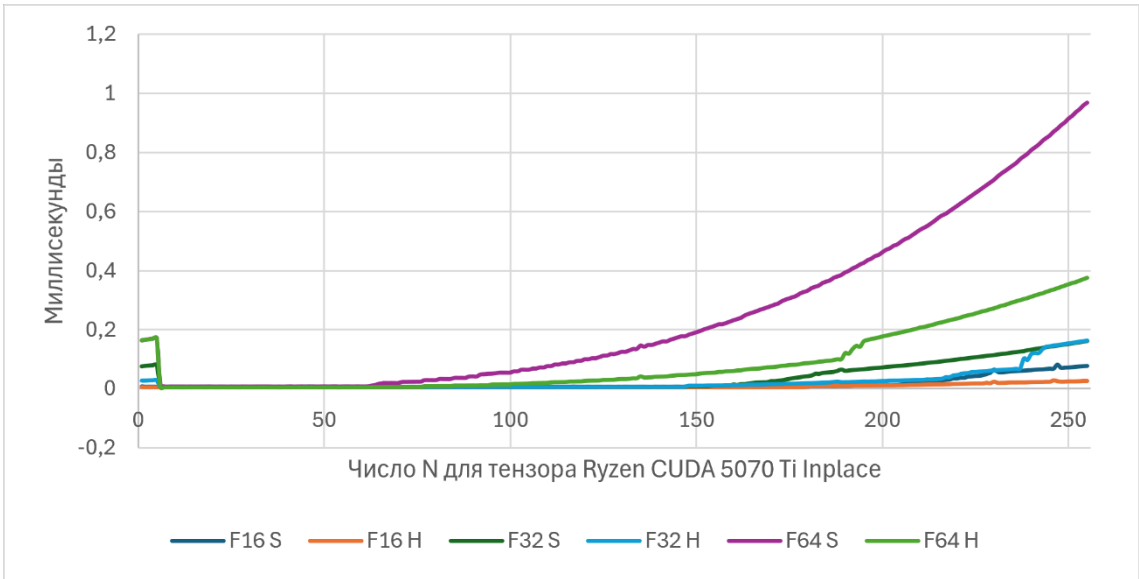


Рисунок 8 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на 5070 Ti с жёсткой сигмоидой Inplace для матриц NxNxN, меньше – лучше

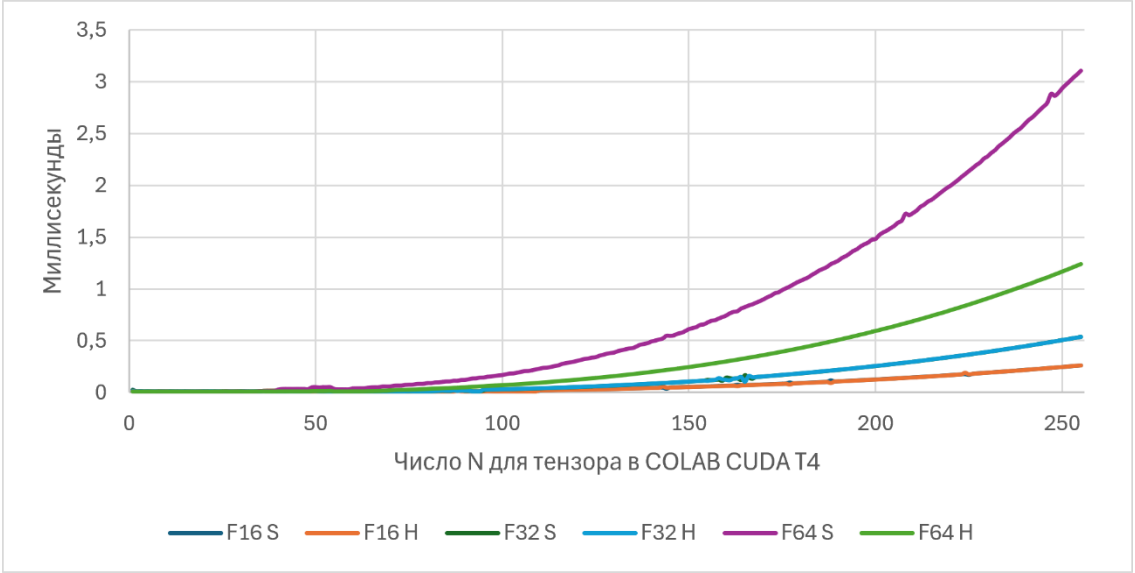


Рисунок 9 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Nvidia Tesla T4 в Google Colab для матриц NxNxN, меньше - лучше

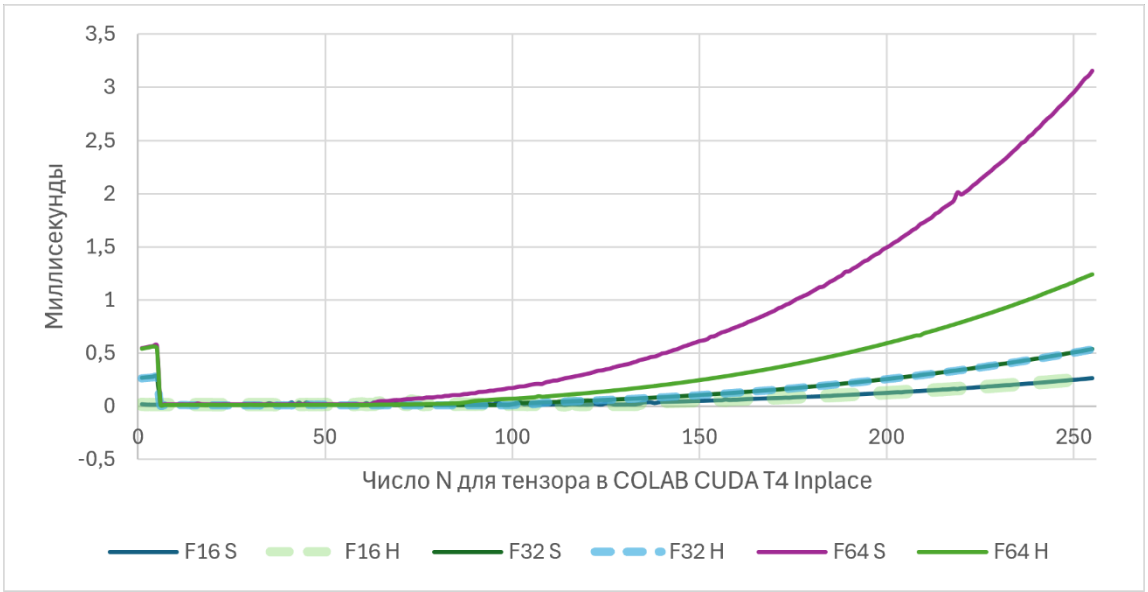


Рисунок 10 – временные затраты в миллисекундах для применения сигмоиды и жёсткой сигмоиды на Nvidia Tesla T4 в Google Colab с жёсткой сигмоидой Inplace для матриц NxNxN, меньше – лучше

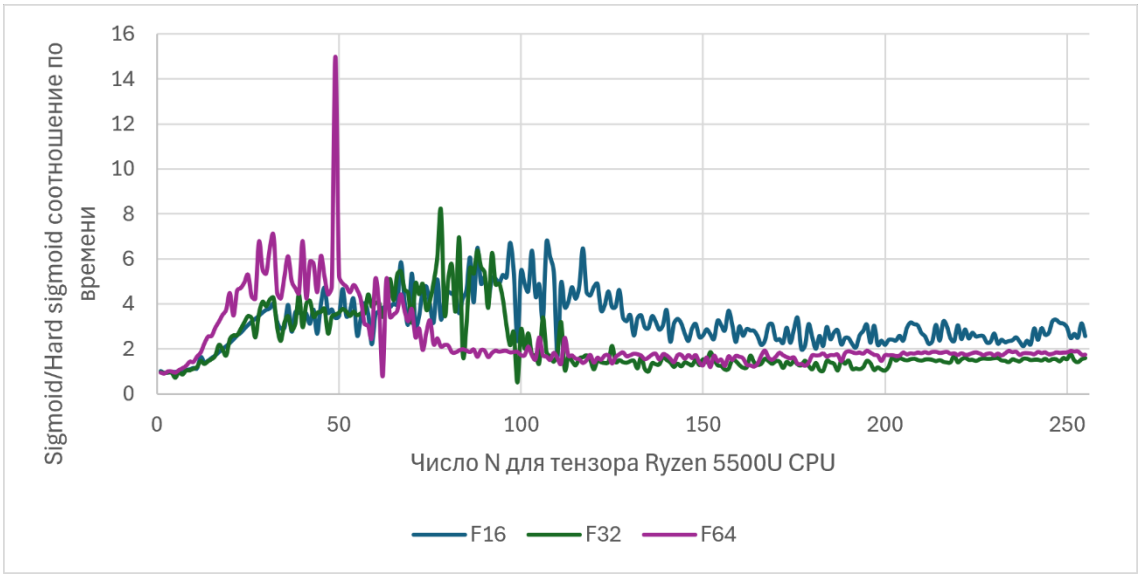


Рисунок 11 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Ryzen 5500U

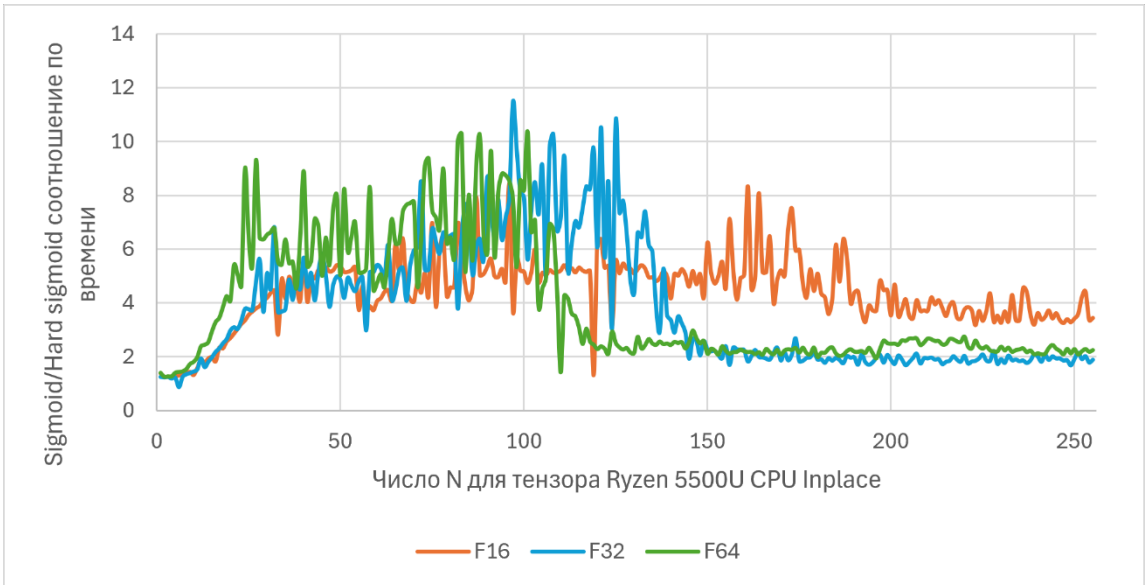


Рисунок 12 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Ryzen 5500U при использовании Inplace

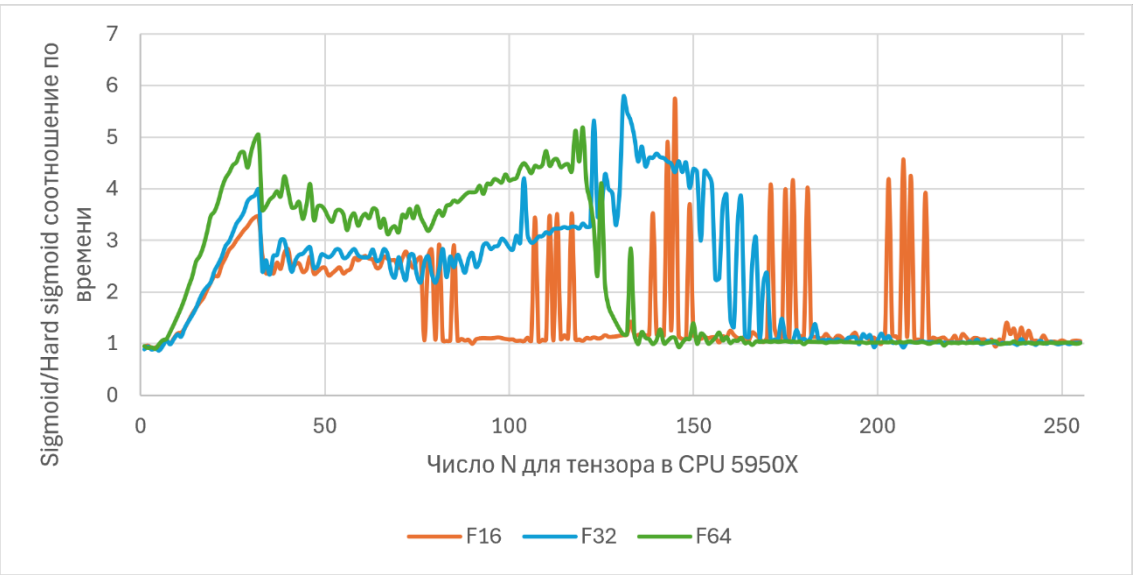


Рисунок 13 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Ryzen 5950X

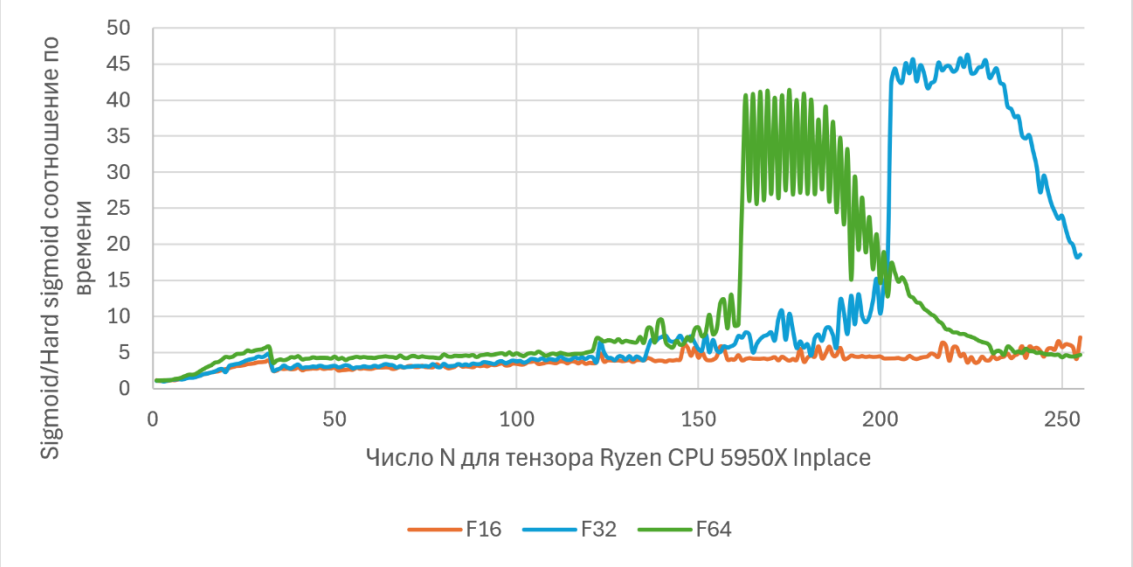


Рисунок 14 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Ryzen 5950X при использовании Inplace

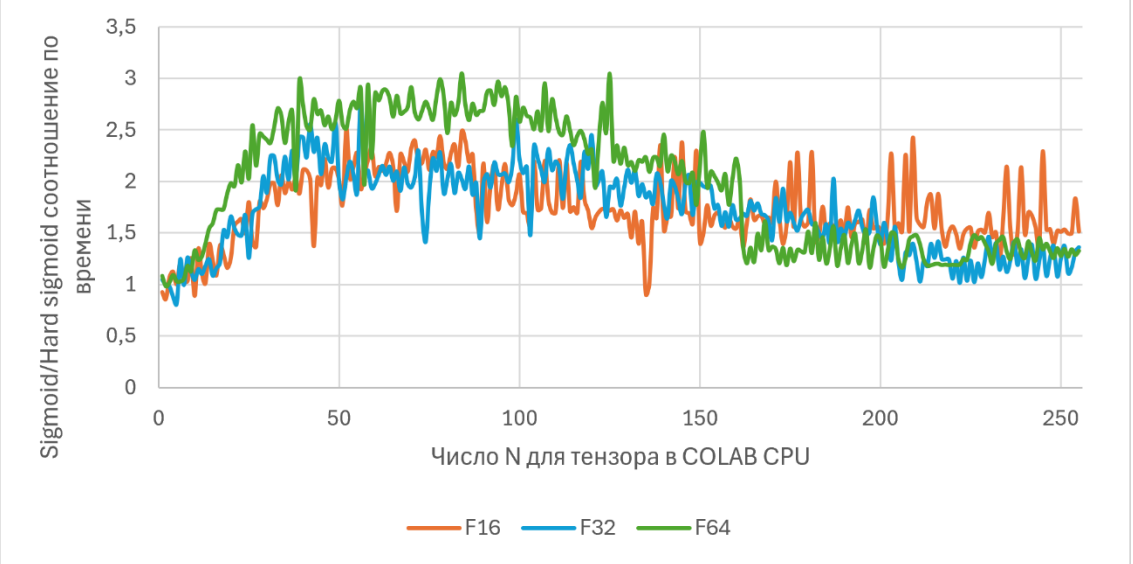
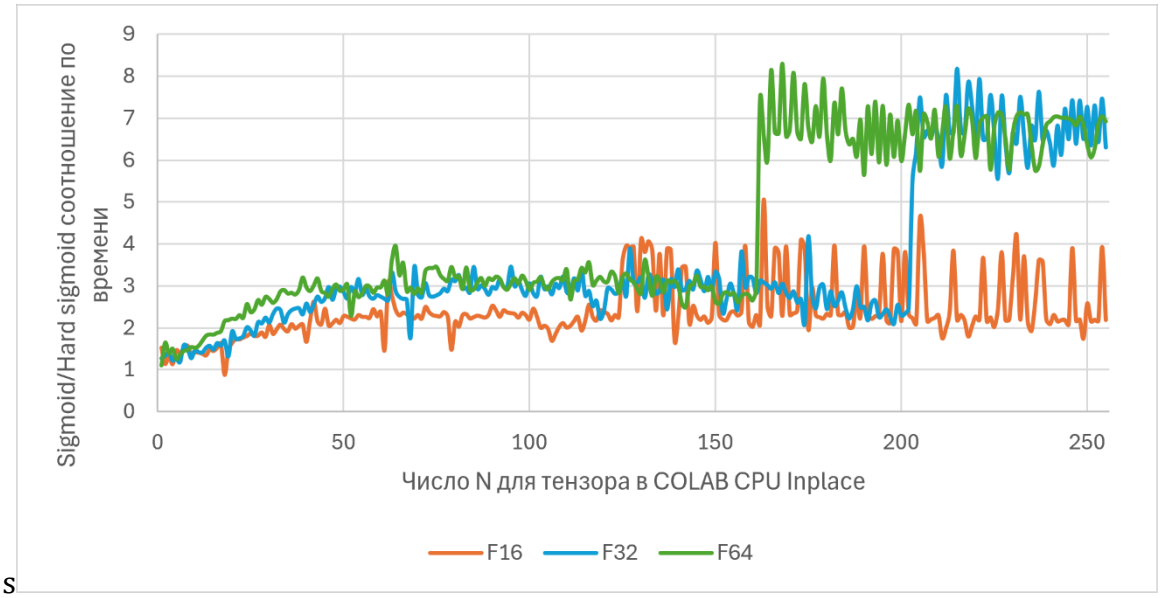


Рисунок 15 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на CPU в Google Colab



s

Рисунок 16 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на CPU в Google Colab при использовании Inplace

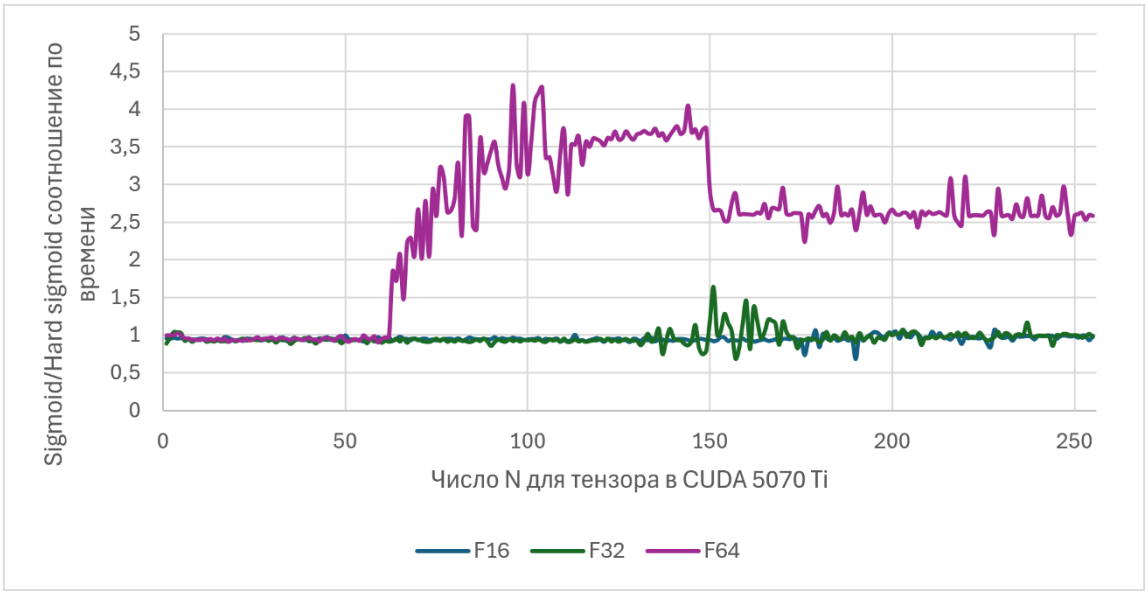


Рисунок 17 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Nvidia RTX 5070 Ti

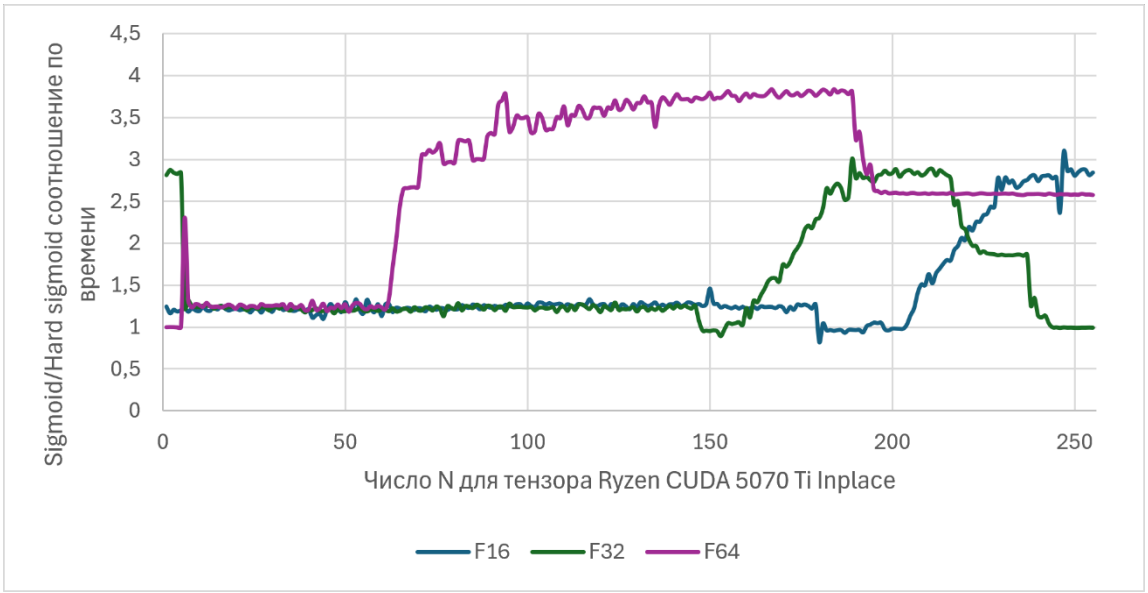


Рисунок 18– отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Nvidia RTX 5070 Ti при использовании Inplace

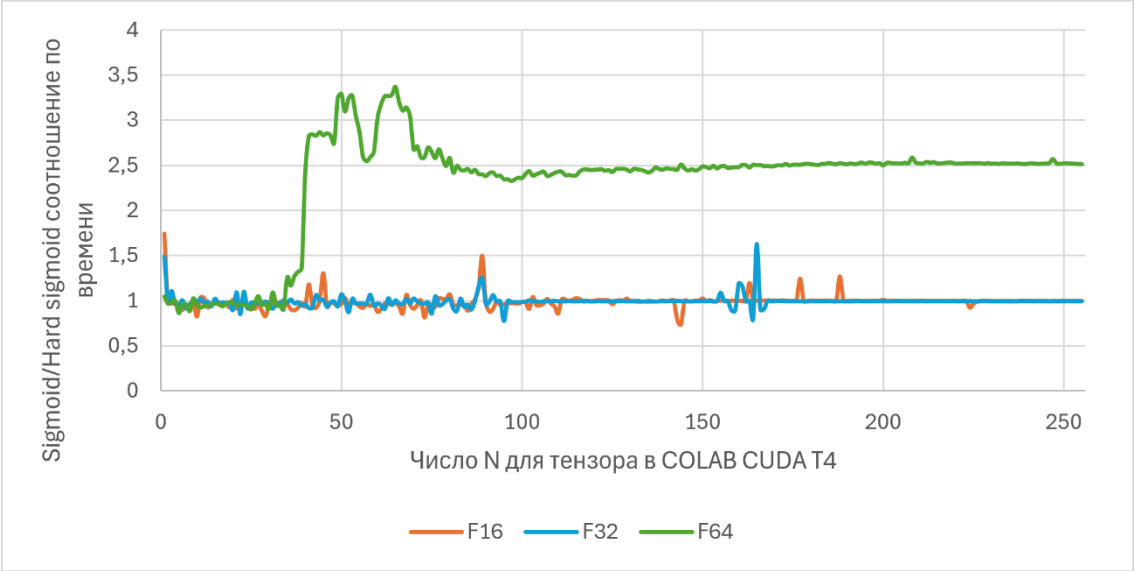


Рисунок 19 – отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Nvidia Tesla T4 в Google Colab

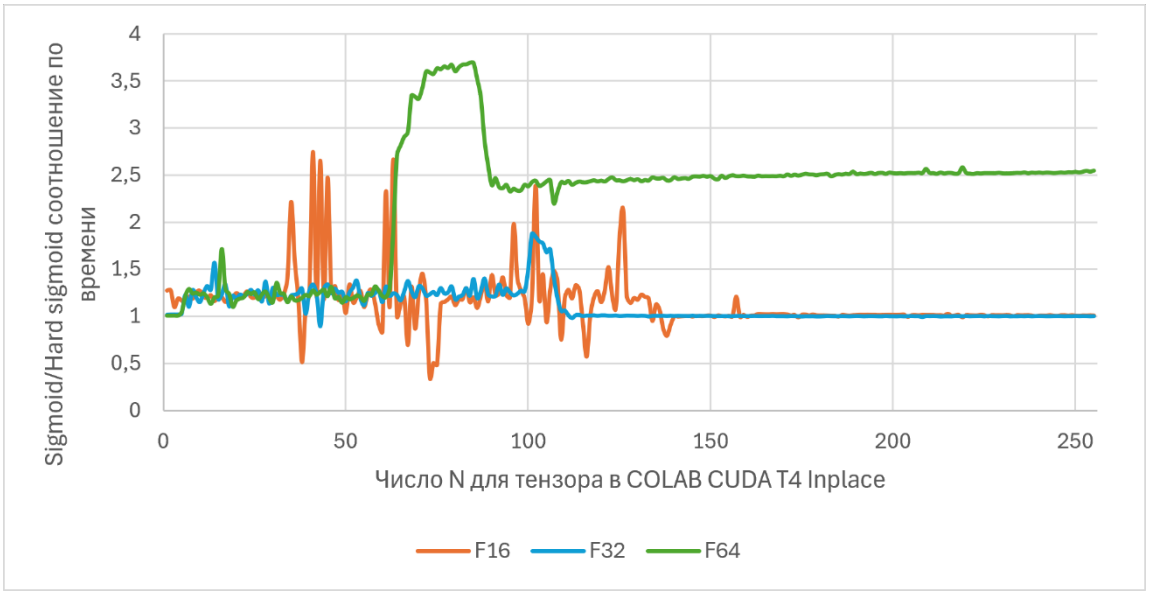


Рисунок 20– отношение временных затрат для применения сигмоиды к временным затратам применения жёсткой сигмоиды на Nvidia Tesla T4 в Google Colab при использовании Inplace

Применение оптимизаций в модуле GRU позволило сократить число параметров модуля с 1614720 до 549248 т.е. в ~2.94 раза.

Изменение в совокупном числе параметров нейросетевой архитектуры DROID-SLAM от исходных 4002565 параметров к 2937093 параметрам т.е. в 1.36 раза или ~27%.

В исходной архитектуре вклад GRU в число параметров составлял ~40.3%, а стал ~18.7% можно утверждать, что полученная архитектура имеет более равномерное распределение по числу параметров по слоям, чем было. Снижение числа требуемых параметров позволило снизить требуемый объём видеопамяти

для хранения матриц внутри модуля GRU и матриц для вычисления градиентов на этапе обучения.

Профилирование памяти CUDA от начала до завершения двух шагов обучения для оригинальным GRU и новым GRU с 8 итерациями и 7 кадрами, показало:

1. Снижение числа запросов на выделение памяти от 29596 до 28932 на ~3%.

2. Снижение максимального потребления от 11.3 Гб до 10.3 Гб VRAM.

3. Наличия большого блока памяти для буфера данных кадров в 864 мегабайта памяти. Размер данного блока зависит от числа кадров. Дополнительное изучение на 1 итерации и 2 кадрах против 8 итераций и 2 кадров показала объём в 72 мегабайт для данного блока.

4. Наличие на вершине потребления памяти операций выделений памяти частями для проективных преобразований, зависящей по числу ступеней от числа итераций. Для числа итераций 8 и числа кадров 7 совокупный объём всех ступеней, связанных с проективными преобразованиями, составил~1.5 Гб.

5. Число уровней пирамиды начинающийся около 2.7 Гб и заканчивающийся около 8.9 Гб по 1-1.3 Гб каждый и зависит от числа итераций. Дополнительно наблюдаются короткосрочные пики, которые могут быть значительно выше среднего уровня потребления памяти на ~500 мегабайт и вносят шум в собранные данные. Дополнительно в этих ступенях прослеживается связь с модулем DROID-SLAM GraphAgg.

Анализ данных показал, что имеются дальнейшие перспективы по оптимизации потребления памяти в направлении: буфера, проективных преобразований, модуля GraphAgg. Совокупное перспективное снижение памяти на 10-20% для дальнейших исследований.

Обучение архитектуры выполнялось на наборе данных, содержащем виртуальные сцены полёта беспилотных летательных аппаратах и робототехнических комплексов в сложных условиях TartanAir [25] на 250000 шагах в несколько этапов.

Для изучения влияния вносимых модификаций выполнено обучение архитектур GRU модуля различных модификаций на одинаковом протоколе обучения:

1. Оригинальный GRU с глобальным контекстом и функцией активации сигмоиды.
2. Оригинальный GRU без глобального контекста и функцией активации сигмоиды.
3. Оригинальный GRU с глобальным контекстом и функцией активации жёсткой сигмоиды.
4. Оригинальный GRU без глобального контекста и функцией активации жёсткой сигмоиды.
5. Новый GRU с глобальным контекстом и функцией активации сигмоиды.
6. Новый GRU без глобального контекста и функцией активации сигмоиды.
7. Новый GRU с глобальным контекстом и функцией активации жёсткой сигмоиды.
8. Новый GRU без глобального контекста и функцией активации жёсткой сигмоиды.

Платформа, на которой выполнялось обучение состояла из: CPU Ryzen 5950X, GPU Nvidia RTX 5070 Ti, 64 Гб RAM 2667 MT/c. Использована операционная система Ubuntu 24.04. Использованы драйвера nvidia-driver-570-open доступные в репозитории graphics-drivers/ppa. Использована версия CUDA 12.8.0, 570.86.10 для Ubuntu 24.04. Вместе с torch 2.10.0 torchvision 0.25.0 torchaudio 2.10.0.

В качестве оптимизатора - Adam, а шедулера для темпа обучения - OneCycleLR, как в оригинальной реализации DROID-SLAM доступной публично [24].

Выполнены замеры, потребления памяти используя функции PyTorch:

1. torch.cuda.device_memory_used – использованная память устройства
2. torch.cuda.memory_allocated – выделенная память CUDA
3. torch.cuda.memory_reserved – зарезервированная память CUDA

Замеры выполнены после 1 шага обучения и вычисления градиентов, и функции потерь для различного числа итераций и числа кадров для оригинального модуля GRU и нового модуля GRU с использованием жёсткой сигмоиды. Для числа

итераций от 1 до 8 включительно и числа кадров от 2 до 7 включительно. В случае, который выбран для исследований экономия составила по зарезервированной памяти 7.23%, по выделенной памяти 14.57%, по использованной памяти устройства 6.37%. Лучший результат по экономии выделенной памяти при 2 кадрах и 1 итерации 21.19%. Лучший результат по экономии зарезервированной памяти и выделенной памяти устройства при 5 кадрах и 7 итерациях составил 28.29% и 23.96% соответственно.

Результаты по потреблению памяти для нового модуля GRU представлены в таблицах: 1, 2, 3 и гистограммах, представленных на рисунках 21, 22, 23.

Результаты по потреблению памяти для оригинального модуля GRU представлены в таблицах: 4, 5, 6 и гистограммах, представленных на рисунках 24, 25, 26.

Сравнение для выбранной комбинации из 7 кадров и 8 итераций представлено на рисунках 27, 28, 29.

Время обучения составило около трёх дней для каждого варианта модификации архитектуры.

Таблица 1

Использованная память устройства на обучении при различном числе итераций и числа кадров для модифицированной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	2717	3725	4481	5619	6811	7595
2	2809	3977	5367	7581	7581	7493
3	2873	4233	5887	8251	9497	9514
4	3056	4724	6457	8607	9039	10901
5	3143	4959	6943	8039	9823	11887
6	3255	5283	7448	10644	12388	12868
7	3366	5416	**7934**	9240	13346	13858
8	**3482**	**5708**	7704	**12206**	**14280**	**14846**

Таблица 2

Выделенная память CUDA на обучении при различном числе итераций и числа кадров для модифицированной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	81	86	88	93	98	100
2	83	88	91	97	97	102
3	85	91	95	102	107	111
4	87	94	99	106	113	118
5	89	96	101	110	117	124
6	91	99	105	114	123	129
7	93	102	108	118	128	136
8	**95**	**104**	**111**	**123**	**134**	**141**

Таблица 3

Зарезервированная память CUDA на обучении при различном числе итераций и числа кадров для модифицированной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	896	1904	2660	3798	4990	5774
2	988	2156	3578	5792	5792	5704
3	1084	2444	4098	6462	7708	7658
4	1200	2868	4602	6752	7184	9046
5	1288	3104	5088	6184	7968	10032
6	1400	3428	5596	8792	10536	11016
7	1514	3564	**6082**	7388	11494	12006
8	**1630**	**3856**	5852	**10354**	**12428**	**12994**

Таблица 4

Использованная память устройства на обучении при различном числе итераций и числа кадров для исходной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	2816	3821	4765	5551	8171	7617
2	2883	4071	5555	7931	7931	9161
3	3007	4393	6153	8443	9659	9869
4	3107	4717	6027	9520	11002	10908
5	3224	5012	6490	9950	12064	11938
6	3328	5322	7144	9070	13120	12968
7	3454	5638	7626	**12152**	12020	14754
8	**3582**	**5952**	**8120**	10416	**15290**	**15856**

Таблица 5

Выделенная память CUDA на обучении при различном числе итераций и числа кадров для исходной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	103	107	109	114	119	122
2	105	110	114	119	119	124
3	107	113	117	123	130	134
4	109	116	120	127	134	139
5	111	118	122	131	139	146
6	113	120	125	135	145	150
7	115	123	130	140	148	158
8	**117**	**126**	**133**	**143**	**154**	**165**

Таблица 6

Зарезервированная память CUDA на обучении при различном числе итераций и числа кадров для исходной архитектуры DROID-SLAM в мегабайтах

Итераций\Кадров	**2**	**3**	**4**	**5**	**6**	**7**
1	924	1960	2904	3690	6310	5756
2	1022	2210	3694	6070	6070	7300
3	1146	2532	4292	6582	7798	8008
4	1246	2856	4166	7670	9152	9058
5	1374	3162	4640	8100	10214	10088
6	1478	3472	5294	7220	11270	11118
7	1604	3788	5776	**10302**	10170	12904
8	**1732**	**4102**	**6270**	8566	**13440**	**14006**

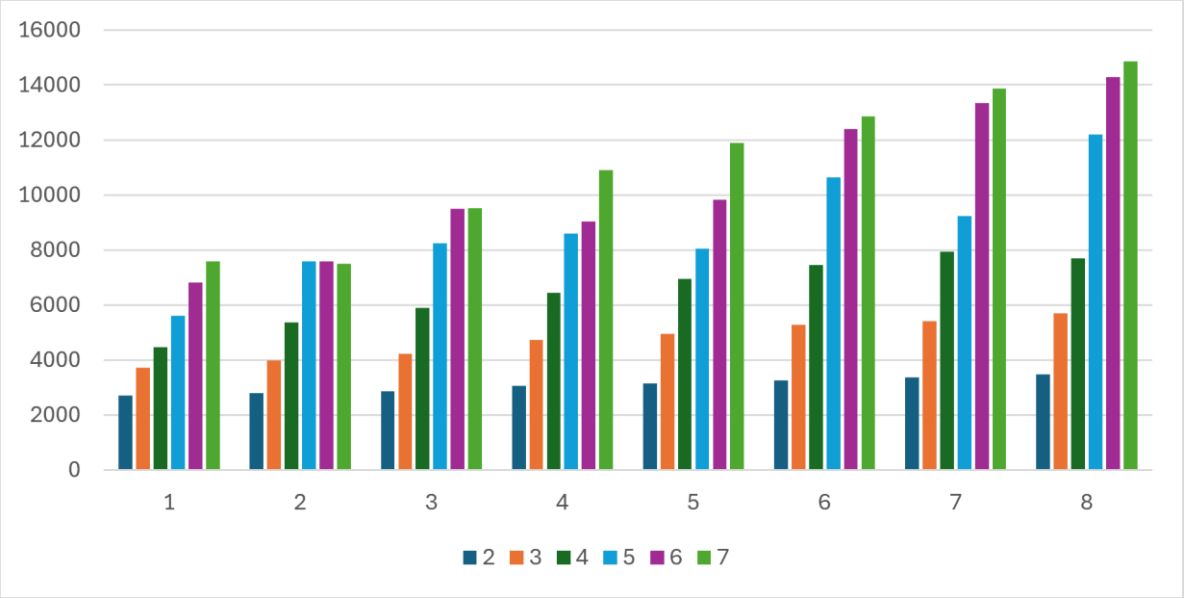


Рисунок 21 – гистограмма использованной памяти устройства в мегабайтах для модифицированной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

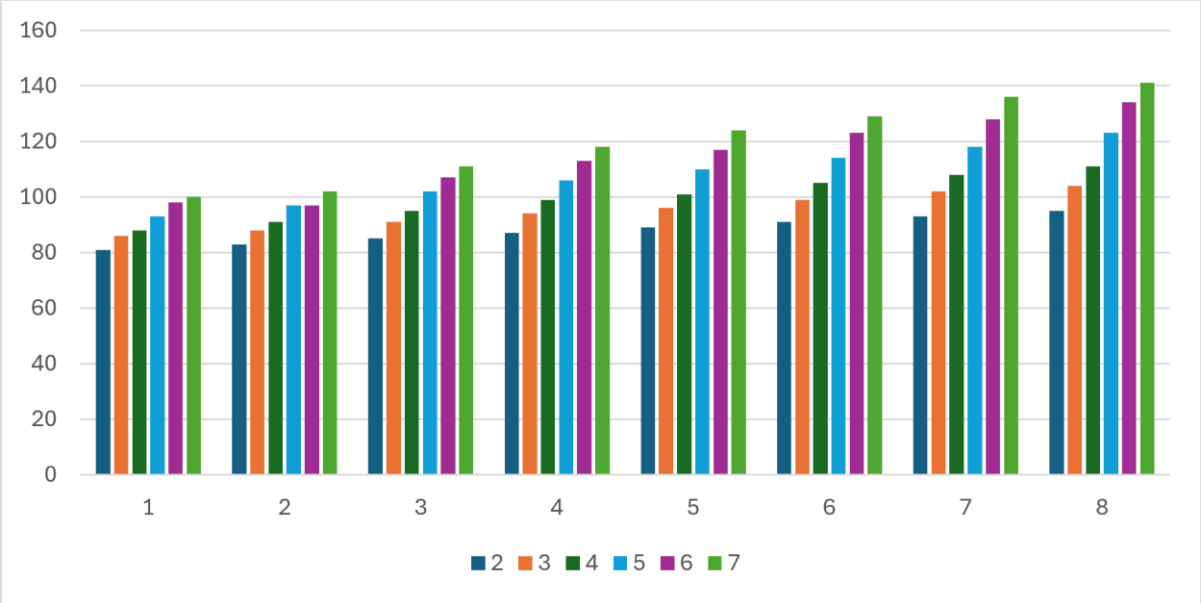


Рисунок 22 – гистограмма выделенной памяти CUDA в мегабайтах для модифицированной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

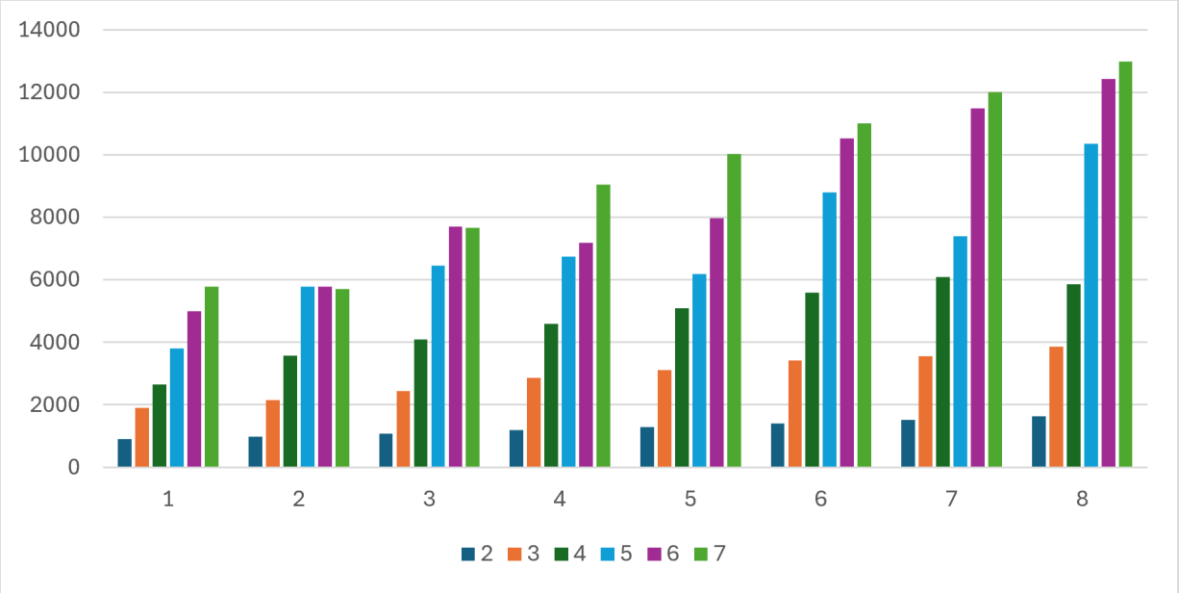


Рисунок 23 – гистограмма зарезервированной памяти CUDA в мегабайтах для модифицированной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

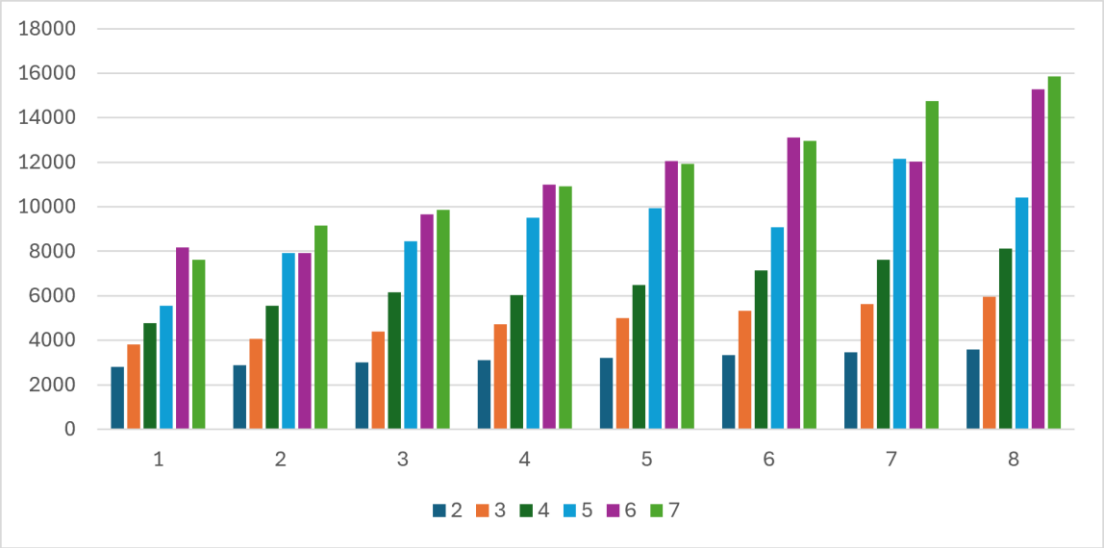


Рисунок 24 – гистограмма использованной памяти устройства в мегабайтах для исходной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

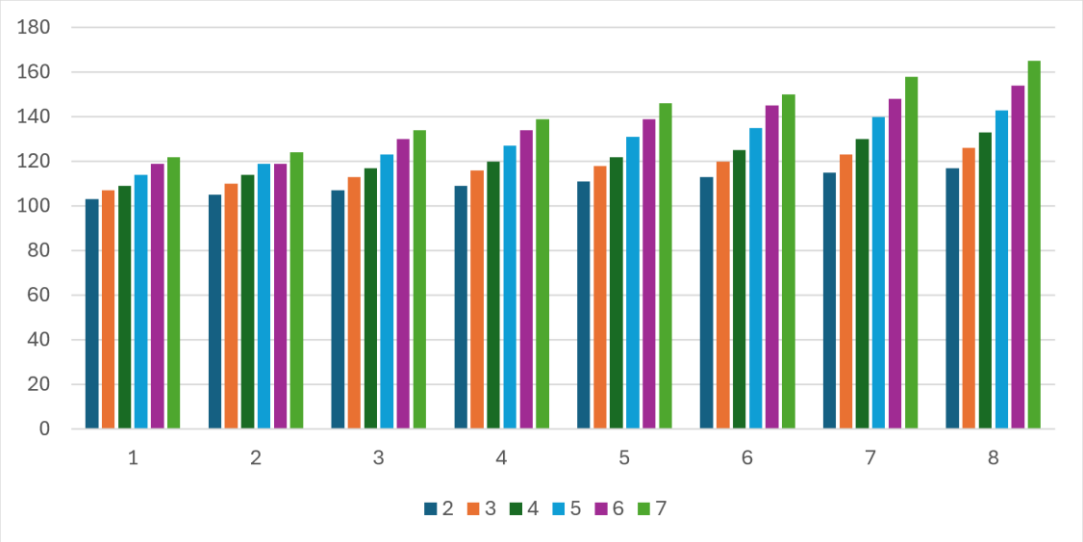


Рисунок 25 – гистограмма выделенной памяти CUDA в мегабайтах для исходной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

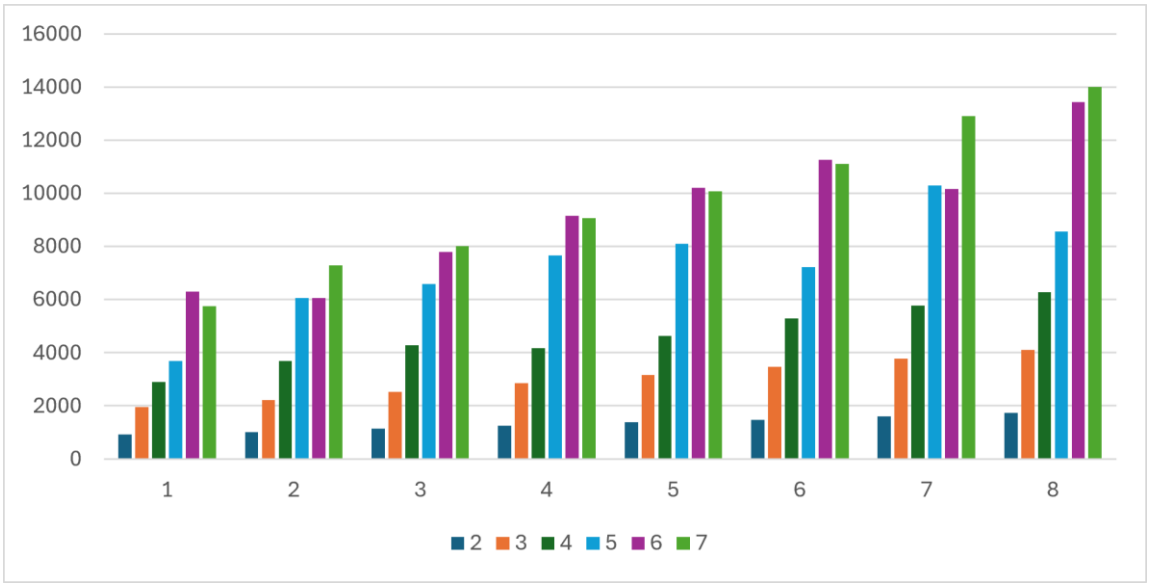


Рисунок 26 – гистограмма зарезервированной памяти CUDA в мегабайтах для исходной архитектуры DROID-SLAM на обучении для числа кадров, сгруппированных по числу итераций

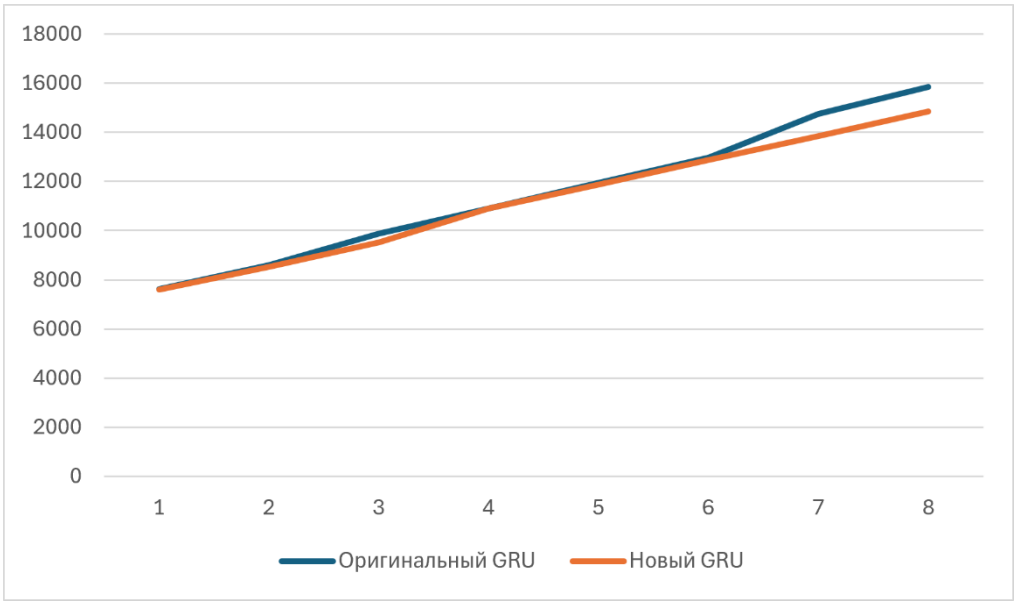


Рисунок 27 – сравнение графиков использованной памяти устройства в мегабайтах на обучении для исходного и модифицированного для 7 кадров и различном числе итераций на обучении

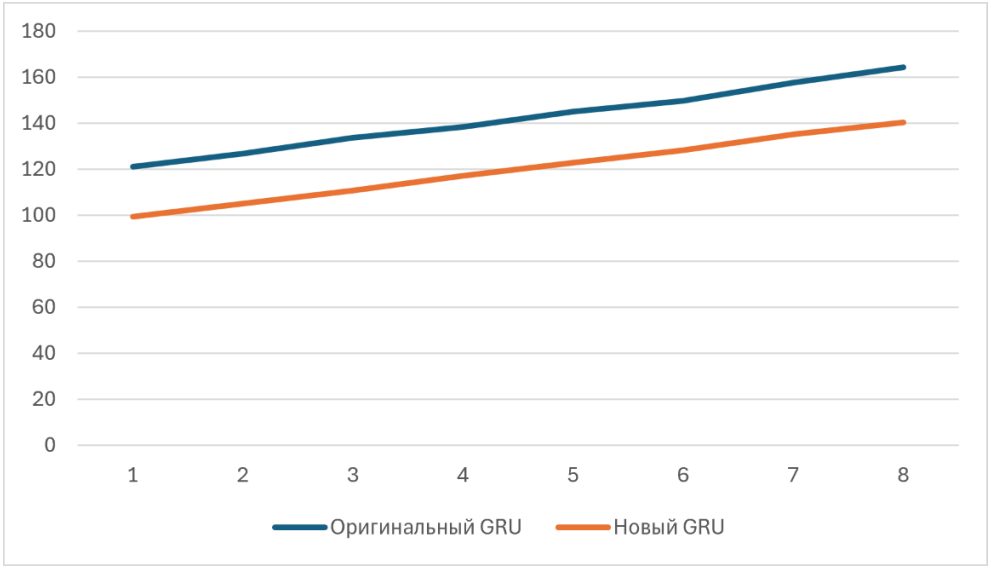


Рисунок 28 – сравнение графиков выделенной памяти CUDA в мегабайтах на обучении для исходного и модифицированного для 7 кадров и различном числе итераций на обучении

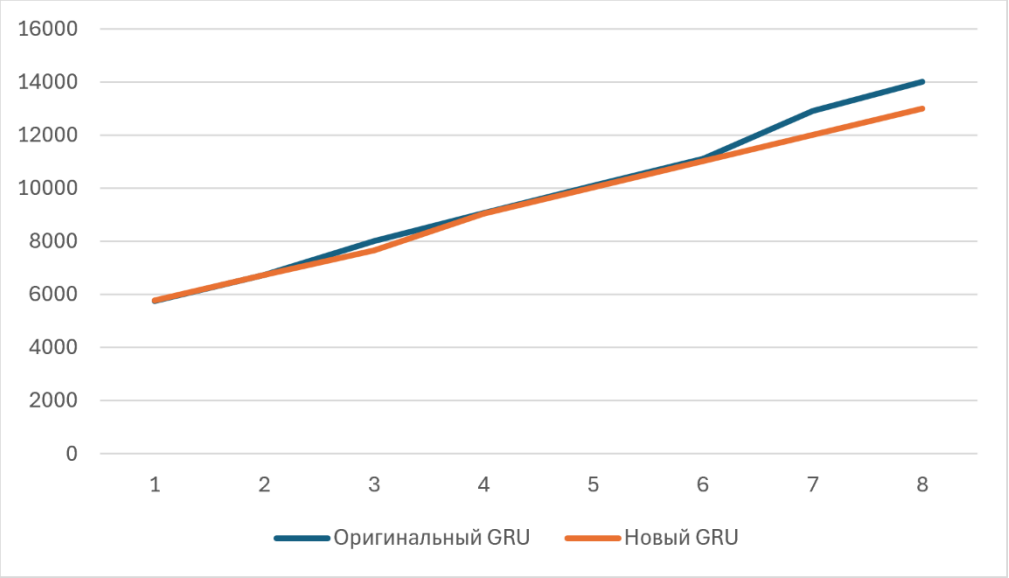


Рисунок 29 – сравнение графиков зарезервированной памяти CUDA в мегабайтах на обучении для исходного и модифицированного для 7 кадров и различном числе итераций на обучении

Для архитектуры DROID-SLAM с оригинальным GRU модулем и с нашим модифицированным модулем GRU на Nvidia RTX 5070 TI и Ryzen 5950X с 64 Гб RAM 2667 MT/c выполнены замеры метрик точности и времени выполнения. Для ориентирования также представлены данные по DROID-SLAM и MINI-DROID-SLAM в таблице из оригинальных статей. При этом, измерения показателей приводим для известных наборов данных EuRoC [19] и TUM [20], а измерения для TartanAir не приводим, по причине того, что показатели не будут репрезентативны из-за участия изображений в обучении. Для датасета EuRoC сцены с префиксом MH (Machine Hall) представляют собой набор чёрно-белых изображений из промышленного машинного зала. Сцены с префиксом V (Vicon's room) содержит в себе набор чёрно-белых изображений комнаты с движущимися объектами и препятствиями. Соответственно сцены градуируются от простых к сложным. MH01, MH02, V101, V202 – низкой сложности, хорошо освещены, хорошие текстуры. MH03, V102, V202 – средней сложности, хорошо освещены, но с быстрым движением. MH04, MH05, V103, V104 – высокой сложности, тёмные сцены и быстрое движение. При этом, важно отметить, что, например у авторов [17] результаты на TartanAir имеют признаки переобучения в силу того, что показатель ошибки положения около нуля на всех сценах.

Данные по замерам собирались, используя 3 прохода. Данные результатов усреднялись и для них вычислялось среднеквадратическое отклонение. Данный подход позволяет понять среднее значение и разброс относительно него.

Референсные значения из статей авторов и нашего модуля GRU с глобальным контекстом и жёсткой сигмоидой представлены в таблицах 7, 8, 9.

Собранные данные для рассматриваемых случаев представлены в таблицах 10-21. В данных таблицах представлены замеры для результатов обучения с одинаковыми настройками по числу шагов, по числу кадров, по числу итераций для архитектуры DROID-SLAM с различными модулями GRU.

1. Оригинальный GRU с сигмоидой представлен в таблицах 10, 11, 12.
2. Оригинальный GRU с жёсткой сигмоидой представлен в таблицах 13, 14, 15.
3. Новый GRU с сигмоидой представлен в таблицах 16, 17, 18.
4. Новый GRU с жёсткой сигмоидой представлен в таблицах 19, 20, 21.

Анализ метрик, собранных по результатам обучения различных модулей GRU, показал следующее:

1. Наблюдается статистически значимое уменьшение времени выполнения у нового модуля GRU до 20% по сравнению с обученным оригинальным.

2. Применение жёсткой сигмоиды в оригинальном GRU оказывает влияние на время выполнения, но из-за имеющегося разброса между результатами нельзя однозначно сказать в какую сторону. Возможно предположить, что снижение точности и получаемое ускорение в выполнении компенсируется дополнительными временными затратами другими модулями визуальной одометрии, например Bundle Adjustment для компенсации ошибки.

3. Применение жёсткой сигмоиды по сравнению с сигмоидой в новом GRU показывает большой разброс по времени выполнения. В среднем наблюдается замедление, но разброс указывает на то, что возможно и ускорение.

4. На собранных данных видно, что убирание глобального контекста в оригинальном GRU позволяет получить снижение ошибки, но не обязательно статистически значимо уменьшает совокупное время выполнения. В новом GRU убирание глобального контекста явным образом снижает точность, но уменьшает время выполнения на дополнительные 5-6%. Предположительно данный

результат возникает в силу того, что слои для получения глобального контекста в новом GRU имеют больший вклад в общее число параметров GRU, чем таковой в оригинальном модуле GRU.

5. Сравнивая оригинальный GRU с новым GRU использующим жёсткую сигмоиду и сигмоиду преимущество по точности в сторону нового GRU с жёсткой сигмоидой и глобальным контекстом.

6. С другой стороны, новый GRU без глобального контекста отстаёт от оригинального GRU. Аналогичные наблюдения для сравнения оригинального GRU с жёсткой сигмоидой при сравнении с новым GRU.

Сравнение с референсными значениями для жёсткой сигмоиды и нового GRU показало следующее:

1. Наблюдается уменьшение времени выполнения у нашего GRU с жёсткой сигмоидой до 20%.

2. По RMSE наше решение близко или превосходит MINI-DROID-SLAM. Но при этом уступает DROID-SLAM.

3. По SSE наше решение уступает DROID-SLAM до 30%.

Таблица 7

Результаты для существующих решений и нашего на наборе данных EuRoC для моно

Метод	**Метрика**	**MH01**	**MH02**	**MH03**	**MH04**	**MH05**	**V101**	**V102**	**V103**	**V201**	**V202**	**V203**
MINI-DROID-SLAM [17]	RMSE м.	0.013	0.014	0.022	0.043	0.043	0.037	0.012	0.020	0.017	0.013	0.014
DROID-SLAM	RMSE м.	0,012	0,011	0,022	0,046	0,043	0,037	0,013	0,019	0,018	0,010	0,014
	SSE $м^2$.	0,526	0,396	1,257	4,239	4,063	3,893	0,267	0,717	0,708	0,211	0,367
	Секунд	131,039	121,396	127,810	64,008	71,533	106,313	122,277	154,762	67,909	159,016	154,369
НОВЫЙ GRU С Г.К.	RMSE м.	0,014	0,013	0,022	0,05	0,043	0,036	0,013	0,019	0,017	0,01	0,015
	SSE $м^2$.	0,766	0,481	1,269	4,983	4,054	3,8	0,285	0,74	0,65	0,242	0,435
	Секунд	113,214	105,227	111,37	53,037	61,317	92,329	106,361	128,39	56,693	132,479	134,761

Таблица 8

Результаты для существующих решений и нашего на наборе данных EuRoC для стерео

Метод	**Метрика**	**MH01**	**MH02**	**MH03**	**MH04**	**MH05**	**V101**	**V102**	**V103**	**V201**	**V202**	**V203**
DROID-SLAM	RMSE м.	0,012	0,012	0,022	0,093	0,041	0,037	0,012	0,019	0,018	0,010	0,055
	SSE $м^2$.	0,564	0,420	1,241	16,965	3,665	3,879	0,234	0,769	0,723	0,210	5,718
	Секунд	154,418	144,701	144,064	75,010	83,290	126,416	130,874	160,471	77,843	163,910	170,808
НОВЫЙ GRU С Г.К.	RMSE м.	0,014	0,014	0,022	0,096	0,042	0,036	0,012	0,019	0,015	0,01	0,054
	SSE $м^2$.	0,718	0,553	1,232	18,295	3,981	3,757	0,257	0,737	0,535	0,238	5,515
	Секунд	133,541	121,552	125,584	65,164	72,846	109,853	116,832	140,44	69,728	145,36	151,63

Таблица 9

Результаты для существующих решений и нашего на наборе данных TUM

Метод	Метрика	360	desk	desk2	floor	plant	room	rpy	reddy	xyz
MINI-DROID-SLAM [6]	RMSE м.	0,111	0,018	0,042	0,021	0,016	0,049	0,026	0,048	0,012
DROID-SLAM	RMSE м.	0,072	0,016	0,026	0,021	0,014	0,049	0,023	0,034	0,010
	SSE м^2.	1,934	0,083	0,218	0,279	0,105	1,668	0,186	0,814	0,043
	Секунд	28,337	21,845	26,773	30,619	37,744	44,512	37,503	50,964	28,769
НОВЫЙ GRU С Г.К.	RMSE м.	0,108	0,018	0,03	0,023	0,017	0,055	0,032	0,035	0,011
	SSE м^2.	4,679	0,104	0,281	0,327	0,166	2,06	0,358	0,872	0,048
	Секунд	26,022	19,406	24,708	29,071	35,404	40,412	35,224	48,432	28,801

Таблица 10

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных EuRoC для моно

	Метрика	**MH01**	**MH02**	**MH03**	**MH04**	**MH05**	**V101**	**V102**	**V103**	**V201**	**V202**	**V203**
ОРИГ. GRU С Г.К.	**RMSE м.**	0,014	0,013	0,874	0,049	0,042	0,037	0,016	0,017	0,019	0,012	0,016
	SSE м^2.	0,696	0,512	5805,921	4,806	3,969	3,893	0,419	0,623	0,776	0,337	0,479
	Секунд	131,986	122,514	125,975	62,491	69,242	107,542	119,901	144,291	66,866	152,939	149,75
	σ(RMSE м.)	0	0	1,202	0	0	0	0	0	0	0	0
	σ(SSE м^2.)	0,001	0	8208,702	0,012	0,022	0,006	0,005	0,001	0,004	0,003	0,004
	σ(Секунд)	1,495	0,373	0,814	0,148	0,408	0,178	0,13	0,275	0,318	0,03	0,957
ОРИГ. GRU БЕЗ Г.К.	**RMSE м.**	0,013	0,012	0,023	0,048	0,044	0,035	0,011	0,018	0,017	0,009	0,015
	SSE м^2.	0,633	0,461	1,384	4,497	4,243	3,581	0,218	0,643	0,669	0,187	0,412
	Секунд	128,436	117,723	125,784	65,62	71,876	109,699	119,395	150,562	67,978	151,666	149,406
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м^2.)	0,001	0,001	0,001	0,031	0,002	0,005	0	0,001	0,001	0,001	0,004
	σ(Секунд)	1,268	0,738	0,608	1,068	1,447	1,892	0,868	2,929	2,036	1,014	0,685

Таблица 11

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных EuRoC для стерео

	Метрика	**MH01**	**MH02**	**MH03**	**MH04**	**MH05**	**V101**	**V102**	**V103**	**V201**	**V202**	**V203**
ОРИГ. GRU С Г.К.	**RMSE м.**	0,014	0,012	0,023	0,097	0,042	0,037	0,015	0,018	0,018	0,012	0,053
	SSE м^2.	0,666	0,46	1,348	18,51	3,96	3,889	0,369	0,685	0,754	0,341	5,235
	Секунд	151,358	137,97	142,313	74,71	80,863	123,421	131,807	157,509	79,222	163,357	166,978
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м^2.)	0	0,001	0,002	0,001	0,002	0,001	0,003	0,002	0,002	0,005	0,001
	σ(Секунд)	0,265	0,074	0,105	0,102	0,144	0,168	0,018	0,323	0,105	0,392	0,414
ОРИГ. GRU БЕЗ Г.К.	**RMSE м.**	0,013	0,012	0,022	0,096	0,042	0,035	0,011	0,018	0,016	0,009	0,054
	SSE м^2.	0,61	0,462	1,303	18,161	3,942	3,571	0,191	0,676	0,563	0,176	5,593
	Секунд	151,068	140,939	143,355	76,413	85,719	126,521	133,139	161,472	80,392	165,478	173,617
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м^2.)	0,001	0,002	0	0,01	0	0,002	0,001	0	0,004	0,001	0
	σ(Секунд)	1,177	1,354	2,782	1,473	2,659	1,485	1,808	2,123	0,744	2,198	3,611

Таблица 12

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных TUM

	Метрика	360	DESK	DESK2	FLOOR	PLANT	ROOM	RPY	TEDDY	XYZ
ОРИГ. GRU С Г.К.	**RMSE м.**	0,146	0,018	0,028	0,023	0,015	0,088	0,029	0,037	0,01
	SSE м².	8,399	0,103	0,252	0,321	0,133	5,246	0,303	0,964	0,043
	Секунд	28,329	21,492	26,684	32,298	39,287	45,119	37,249	53,944	31,59
	σ(RMSE м.)	0,028	0	0	0	0	0,006	0	0	0
	σ(SSE м².)	3,243	0	0	0	0,005	0,704	0,004	0,027	0
	σ(Секунд)	0,258	0,041	0,047	0,16	0,148	0,137	0,11	0,134	0,207
ОРИГ. GRU БЕЗ Г.К.	**RMSE м.**	0,117	0,018	0,03	0,021	0,019	0,059	0,028	0,037	0,011
	SSE м².	5,144	0,102	0,275	0,274	0,212	2,397	0,276	0,954	0,048
	Секунд	29,78	21,792	27,868	33,162	40,157	45,934	39,29	53,738	31,027
	σ(RMSE м.)	0,003	0	0	0	0	0	0,001	0	0
	σ(SSE м².)	0,242	0	0	0	0,004	0,004	0,027	0,004	0
	σ(Секунд)	0,608	0,235	0,317	0,357	0,435	0,534	1,33	1,003	0,681

Таблица 13

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных EuRoC для моно и жёсткой сигмоидой

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
ОРИГ. GRU С Г.К.	**RMSE м.**	0,014	0,013	0,023	0,05	0,042	0,037	0,016	0,017	0,018	0,012	0,016
	SSE м².	0,733	0,509	1,365	4,942	3,854	3,894	0,416	0,613	0,764	0,323	0,514
	Секунд	132,291	123,81	128,372	63,009	69,666	108,1	120,665	145,571	66,789	152,309	148,811
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,003	0,001	0,026	0,008	0,011	0,013	0,004	0,001	0,009	0,004	0,005
	σ(Секунд)	0,57	0,062	0,485	0,135	0,79	0,271	0,265	0,245	0,343	1,752	2,306
ОРИГ. GRU БЕЗ Г.К.	**RMSE м.**	0,014	0,013	0,023	0,048	0,044	0,035	0,011	0,018	0,018	0,009	0,015
	SSE м².	0,668	0,496	1,387	4,476	4,311	3,596	0,217	0,649	0,683	0,176	0,422
	Секунд	127,673	118,12	124,211	63,038	68,997	106,3	118,546	145,372	64,853	149,826	145,231
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,001	0,005	0,006	0,071	0,036	0,002	0	0,001	0,002	0,001	0,003
	σ(Секунд)	0,702	0,451	0,129	0,186	0,197	0,488	0,702	0,568	0,18	0,413	1,096

Таблица 14

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных EuRoC для стерео и жёсткой сигмоидой

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
ОРИГ. GRU С Г.К.	RMSE м.	0,014	0,013	0,022	0,097	0,043	0,037	0,015	0,018	0,018	0,012	0,053
	SSE м².	0,683	0,473	1,321	18,478	4,041	3,918	0,365	0,693	0,739	0,336	5,316
	Секунд	153,55	140,143	144,28	75,683	81,808	125,301	132,122	158,649	80,198	165,886	171,582
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,001	0	0	0,007	0,015	0,002	0,003	0,003	0,004	0,003	0,012
	σ(Секунд)	0,094	0,141	0,216	0,063	0,431	0,117	0,161	0,459	0,053	0,389	0,31
ОРИГ. GRU БЕЗ Г.К.	RMSE м.	0,012	0,013	0,022	0,096	0,043	0,035	0,011	0,018	0,016	0,009	0,054
	SSE м².	0,572	0,5	1,313	18,2	4,089	3,579	0,195	0,682	0,558	0,186	5,624
	Секунд	147,697	138,724	140,543	74,342	82,302	124,651	130,699	157,567	79,007	163,237	170,302
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,003	0,001	0,004	0,027	0,004	0,004	0	0,001	0,006	0	0,003
	σ(Секунд)	0,176	0,168	0,516	0,232	0,35	0,383	0,164	0,468	0,056	0,281	0,597

Таблица 15

Результаты для обученного оригинального GRU с глобальным контекстом и без на наборе данных TUM и жёсткой сигмоидой

	Метрика	360	DESK	DESK2	FLOOR	PLANT	ROOM	RPY	TEDDY	XYZ
ОРИГ. GRU С Г.К.	RMSE м.	0,087	0,019	0,029	0,023	0,015	0,817	0,032	0,035	0,01
	SSE м².	2,942	0,106	0,265	0,322	0,127	454,256	0,368	0,876	0,043
	Секунд	28,849	21,746	26,564	32,361	39,401	43,146	37,956	53,597	31,543
	σ(RMSE м.)	0,017	0	0	0	0	0	0	0	0
	σ(SSE м².)	1,162	0	0	0,001	0,001	0,526	0,008	0,014	0
	σ(Секунд)	0,051	0,027	0,015	0,066	0,208	0,01	0,137	0,227	0,164
ОРИГ. GRU БЕЗ Г.К.	RMSE м.	0,135	0,018	0,03	0,022	0,019	0,064	0,025	0,036	0,011
	SSE м².	6,857	0,096	0,274	0,284	0,197	2,824	0,228	0,916	0,048
	Секунд	28,566	21,073	26,808	32,104	38,789	44,265	36,762	51,868	29,746
	σ(RMSE м.)	0,006	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,622	0	0	0,001	0,005	0,007	0,003	0,008	0
	σ(Секунд)	0,273	0,07	0,069	0,081	0,009	0,082	0,087	0,15	0,085

Таблица 16

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных EuRoC для моно

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,013	0,012	0,022	0,05	0,043	0,037	0,013	0,02	0,018	0,01	0,015
	SSE м².	0,649	0,442	1,304	4,843	4,052	3,838	0,281	0,808	0,71	0,231	0,44
	Секунд	112,257	104,289	109,488	53,119	60,451	92,241	106,84	128,8	56,605	132,397	137,479
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,009	0,002	0,004	0,022	0,012	0,001	0,001	0,001	0,006	0,001	0,001
	σ(Секунд)	1,679	0,167	0,16	0,13	0,092	0,137	0,425	0,399	0,346	0,176	1,145
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,017	0,017	0,026	0,057	0,053	0,802	0,012	0,023	0,018	0,022	0,018
	SSE м².	1,014	0,853	1,806	6,394	6,23	1884,846	0,23	1,149	0,734	1,067	0,598
	Секунд	110,526	101,626	107,322	52,305	57,198	90,236	99,992	122,22	53,519	127,173	125,258
	σ(RMSE м.)	0	0	0	0	0	0,113	0	0	0	0	0
	σ(SSE м².)	0,005	0,014	0,003	0,006	0,005	533,952	0,002	0,003	0,005	0,01	0,005
	σ(Секунд)	0,354	0,288	0,306	0,165	0,254	0,319	0,686	0,514	0,452	0,381	1,629

Таблица 17

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных EuRoC для стерео

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,014	0,013	0,022	0,097	0,043	0,036	0,012	0,02	0,016	0,01	0,054
	SSE м².	0,678	0,512	1,279	18,723	4,039	3,784	0,235	0,797	0,573	0,237	5,428
	Секунд	131,467	121,264	124,729	64,904	72,56	109,047	116,399	141,158	69,43	144,441	148,18
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0	0	0	0,001	0,004	0,003	0	0,004	0,004	0,002	0,011
	σ(Секунд)	0,144	0,267	0,127	0,256	0,411	0,305	0,088	0,143	0,155	0,639	1,286
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,015	0,014	0,025	0,101	0,053	0,036	0,011	0,02	0,016	0,014	0,057
	SSE м².	0,857	0,622	1,662	20,284	6,261	3,62	0,209	0,809	0,562	0,484	6,116
	Секунд	130,935	118,596	122,252	63,27	69,024	106,418	115,142	136,95	66,458	139,358	146,972
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0	0,001	0,001	0,002	0,016	0,008	0	0	0,001	0,003	0,005
	σ(Секунд)	0,354	0,165	0,133	0,343	0,248	0,073	0,097	0,07	0,045	0,228	0,451

Таблица 18

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных TUM

	Метрика	360	DESK	DESK2	FLOOR	PLANT	ROOM	RPY	TEDDY	XYZ
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,084	0,018	0,028	0,023	0,017	0,055	0,024	0,035	0,011
	SSE м².	2,718	0,103	0,256	0,331	0,162	2,09	0,214	0,854	0,049
	Секунд	26,144	19,724	25,031	28,512	35,412	40,688	36,108	47,886	29,304
	σ(RMSE м.)	0,008	0	0	0	0	0	0	0,001	0
	σ(SSE м².)	0,535	0	0,001	0	0,01	0,01	0,001	0,042	0
	σ(Секунд)	0,054	0,028	0,06	0,156	0,121	0,095	0,103	0,199	0,223
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,174	0,019	0,024	0,026	0,025	0,201	0,039	0,044	0,011
	SSE м².	11,56	0,106	0,187	0,408	0,364	27,478	0,536	1,387	0,05
	Секунд	25,522	18,733	23,948	27,351	33,584	38,929	34,012	46,326	27,515
	σ(RMSE м.)	0,014	0	0	0	0	0,001	0	0,003	0
	σ(SSE м².)	1,908	0	0	0,001	0,002	0,347	0,001	0,17	0
	σ(Секунд)	0,4	0,001	0,097	0,056	0,069	0,086	0,063	0,101	0,12

Таблица 19

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных EuRoC для моно и жёсткой сигмоидой

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,014	0,013	0,022	0,05	0,043	0,036	0,013	0,019	0,017	0,01	0,015
	SSE м².	0,766	0,481	1,269	4,983	4,054	3,8	0,285	0,74	0,65	0,242	0,435
	Секунд	113,214	105,227	111,37	53,037	61,317	92,329	106,361	128,39	56,693	132,479	134,761
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,002	0,01	0,001	0,019	0,019	0	0,004	0,002	0,005	0,001	0,012
	σ(Секунд)	0,817	0,036	0,13	0,031	0,14	0,618	0,216	0,523	0,124	0,125	1,778
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,017	0,016	0,028	0,058	0,057	0,231	0,013	0,02	0,019	0,02	0,018
	SSE м².	1,034	0,738	1,992	6,6	7,171	271,866	0,287	0,832	0,808	0,944	0,609
	Секунд	117,107	105,922	113,882	54,552	58,737	92,75	103,499	125,406	54,133	129,048	126,468
	σ(RMSE м.)	0	0	0	0	0	0,204	0	0	0	0,001	0
	σ(SSE м².)	0,002	0,024	0,012	0,075	0,016	340,4	0,001	0,001	0,01	0,069	0,004
	σ(Секунд)	6,544	5,715	5,604	2,784	3,18	3,26	3,915	0,826	0,297	0,38	1,314

Таблица 20

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных EuRoC для стерео и жёсткой сигмоидой

	Метрика	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,014	0,014	0,022	0,096	0,042	0,036	0,012	0,019	0,015	0,01	0,054
	SSE м².	0,718	0,553	1,232	18,295	3,981	3,757	0,257	0,737	0,535	0,238	5,515
	Секунд	133,541	121,552	125,584	65,164	72,846	109,853	116,832	140,44	69,728	145,36	151,63
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0	0,002	0,001	0,01	0,001	0,001	0,001	0,002	0,001	0,001	0,04
	σ(Секунд)	0,172	0,237	0,105	0,275	0,332	0,415	0,118	0,608	0,308	0,197	0,492
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,016	0,015	0,026	0,101	0,056	0,036	0,012	0,02	0,017	0,016	0,056
	SSE м².	0,9	0,679	1,716	20,087	7,081	3,725	0,253	0,883	0,615	0,613	5,837
	Секунд	131,883	121,021	127,696	66,01	71,852	110,627	119,356	141,485	69,547	145,684	150,394
	σ(RMSE м.)	0	0	0	0	0	0	0	0	0	0	0
	σ(SSE м².)	0,004	0,004	0,005	0,012	0,011	0,003	0	0,001	0,001	0,002	0,03
	σ(Секунд)	0,294	6,023	6,649	3,562	4,067	5,643	5,991	6,816	3,635	7,462	6,072

Таблица 21

Результаты для обученного нового GRU с глобальным контекстом и без на наборе данных TUM и жёсткой сигмоидой

	Метрика	360	DESK	DESK2	FLOOR	PLANT	ROOM	RPY	TEDDY	XYZ
НОВЫЙ GRU С Г.К.	**RMSE м.**	0,108	0,018	0,03	0,023	0,017	0,055	0,032	0,035	0,011
	SSE м².	4,679	0,104	0,281	0,327	0,166	2,06	0,358	0,872	0,048
	Секунд	26,022	19,406	24,708	29,071	35,404	40,412	35,224	48,432	28,801
	σ(RMSE м.)	0,028	0	0	0	0	0	0	0	0
	σ(SSE м².)	2,52	0	0	0,001	0,003	0,006	0	0,014	0
	σ(Секунд)	0,153	0,017	0,059	0,06	0,044	0,124	0,063	0,171	0,128
НОВЫЙ GRU БЕЗ Г.К.	**RMSE м.**	0,199	0,019	0,026	0,03	0,28	0,534	0,04	0,037	0,011
	SSE м².	14,917	0,11	0,22	0,529	91,741	193,89	0,56	0,973	0,047
	Секунд	26,36	19,48	24,764	28,681	33,094	39,488	35,165	48,88	28,349
	σ(RMSE м.)	0,001	0	0	0	0,289	0,004	0	0	0
	σ(SSE м².)	0,172	0	0	0,001	124,467	2,62	0,005	0,009	0
	σ(Секунд)	1,895	1,087	1,336	1,654	4,267	2,278	2,029	2,331	1,773

Заключение

1. Предложенный подход к оптимизации нейросетевого модуля GRU позволил снизить чисто параметров в архитектуре DROID-SLAM на ~27% от исходных 4002565 параметров к 2937093. Снижение числа параметров самого модуля GRU находящегося в составе архитектуре DROID-SLAM составило ~66%, то есть с 1614720 до 549248.

2. Временные затраты при выполнении архитектуры DROID-SLAM снизились в предельном случае на ~20% относительно исходной архитектуры DROID-SLAM на платформе, состоящей из Ryzen 5950X, Nvidia RTX 5070 Ti и 64 Гб RAM.

3. Исследование времени необходимого на применение жёсткой сигмоиды на процессоре Ryzen 5500U и Ryzen 5950X по сравнению с сигмоидой показало перспективу использования жёсткой сигмоиды на процессоре.

4. Исследование времени необходимого на применение жёсткой сигмоиды на видеоадаптере Nvidia RTX 5070 Ti и Tesla T4 по сравнению с сигмоидой показало перспективу использования жёсткой сигмоиды на видеоадаптере при условии применения Inplace варианта функции.

5. В исходной архитектуре вклад GRU в число параметров составлял ~40.3%, а стал ~18.7% в результате чего возможно утверждать, что полученная архитектура имеет более равномерное распределение числа параметров по слоям, чем было в исходной архитектуре.

6. В случае, который выбран для исследований с 7 кадрами и 8 итерациями экономия составила по зарезервированной памяти 7.23%, по выделенной памяти 14.57%, по использованной памяти устройства 6.37%.

7. По сравнению с аналогами (например, MINI-DROID-SLAM) предложенный подход сохраняет дополнительную свёртку и глобальный контекст, что позволяет достичь лучшего результата.

8. Проведены подробные замеры на имеющейся аппаратной платформе для сравнения с существующей архитектурой DROID-SLAM.

В результате выполнения работы по оптимизации модуля GRU, получен вариант оптимизированного модуля GRU с свёртками, показавшего применимость в задаче визуальной одометрии в существующей архитектуре DROID-SLAM. При этом, предложенные решения по оптимизации систем визуальной одометрии актуальны и для наземной робототехники, где также имеются ограничения на доступные вычислительные ресурсы. Можно ожидать, что преимущество по скорости сохранится, а точность может варьироваться в зависимости от характера движения (например, длительные прямолинейные участки со слаботекстурированными поверхностями). Проверка на датасетах для наземных роботов (KITTI [26], Oxford RobotCar [27]) представляет перспективу для дальнейших исследований.

Конфликт интересов

Автор заявляет об отсутствии конфликта интересов.

Conflict of interest

The author declares no conflict of interest.

Список источников

1. Дорошев А.С., Шеломанов Д.А. Методика подбора гиперпараметров нейросетевой модели в задачах оптической навигации // Труды МАИ. 2025. № 142. URL: https://trudymai.ru/published.php?ID=185106

2. The $7 trillion data center build-out: How industrials can capture their share // https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-7-trillion-dollar-data-center-build-out-how-industrials-can-capture-their-share (дата обращения 24.05.2026)

3. Global RAM shortage appears set to continue through 2027 // https://www.computerworld.com/article/4161043/the-memory-shortage-appears-set-to-continue-through-2027.html (дата обращения 24.05.2026)

4. Data centers will consume 70 percent of memory chips made in 2026 - supply shortfall will cause the chip shortage to spread to other segments // https://www.tomshardware.com/pc-components/ram/data-centers-will-consume-70-percent-of-memory-chips-made-in-2026-supply-shortfall-will-cause-the-chip-shortage-to-spread-to-other-segments (дата обращения 30.05.2026)

5. Корыткин П.Г. Применение глубоких нейронных сетей в задаче визуальной одометрии // Труды МАИ. 2025. № 145. URL: https://trudymai.ru/published.php?ID=186898

6. Sen Wang, Ronald Clark, Hongkai Wen, Niki Trigoni // DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 2043–50. https://doi.org/10.1109/icra.2017.7989236

7. Olivier Brochu Dufour and Abolfazl Mohebbi and Sofiane Achiche, An Attention-Based Deep Learning Architecture for Real-Time Monocular Visual Odometry: Applications to GPS-free Drone Navigation, 2024 https://arxiv.org/abs/2404.17745

8. Teed, Zachary and Deng, Jia, “DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras”, Advances in neural information processing systems, 2021.

9. Tabassum, T.E.; Xu, Z.; Petrunin, I.; Rana, Z.A. Integrating GRU with a Kalman Filter to Enhance Visual Inertial Odometry Performance in Complex Environments. Aerospace 2023, 10, 923. https://doi.org/10.3390/aerospace10110923

10. J. C. Heck and F. M. Salem, "Simplified minimal gated unit variations for recurrent neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1593-1596, doi: 10.1109/MWSCAS.2017.8053242.

11. Moyang Li, Zihan Zhu, Marc Pollefeys, Daniel Barath // DROID-SLAM in the Wild. 2026 https://arxiv.org/abs/2603.19076

12. Y. Huang, L. Ji, J. Sun, X. Jiang and H. Liu, "CUEMP: Correspondence Uncertainty Estimation With Motion Priors for Dense Visual Odometry," in IEEE Robotics and Automation Letters, vol. 11, no. 4, pp. 4497-4504, April 2026, doi: 10.1109/LRA.2026.3666386.

13. Roberto Pellerito, Marco Cannici, Daniel Gehrig, Joris Belhadj, Olivier Dubois-Matra, Massimo Casasco, Davide Scaramuzza, “End-to-end Learned Visual Odometry with Events and Frames”, 2024, https://arxiv.org/abs/2309.09947

14. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, Max Tegmark // KAN: Kolmogorov-Arnold Networks, 2024 https://arxiv.org/abs/2404.19756

15. Yesmine Abdennadher, Eleonora Cicciarella, Michele Rossi // Convolutional Spiking-based GRU Cell for Spatio-temporal Data, 2025 IEEE 35th International Workshop on Machine Learning for Signal Processing (MLSP), 2025 http://dx.doi.org/10.1109/MLSP62443.2025.11204318

16. Mateos-Ramirez, P.; Gomez-Avila, J.; Villaseñor, C.; Arana-Daniel, N. Visual Odometry in GPS-Denied Zones for Fixed-Wing Unmanned Aerial Vehicle with Reduced

Accumulative Error Based on Satellite Imagery. Appl. Sci. 2024, 14, 7420. https://doi.org/10.3390/app14167420

17. Albukhari, I.; El-Sayed, A.; Alshibli, M. MINI-DROID-SLAM: Improving Monocular Visual SLAM Using MINI-GRU RNN Network. Sensors 2025, 25, 5448. https://doi.org/10.3390/s25175448

18. Grupp, Michael. evo: Python package for the evaluation of odometry and SLAM., 2017 https://github.com/MichaelGrupp/evo

19. Burri, Michael and Nikolic, Janosch and Gohl, Pascal and Schneider, Thomas and Rehder, Joern and Omari, Sammy and Achtelik, Markus W and Siegwart, Roland // The EuRoC micro aerial vehicle datasets, 2016 https://doi.org/10.1177/0278364915620033

20. J. Sturm and N. Engelhard and F. Endres and W. Burgard and D. Cremers // A Benchmark for the Evaluation of RGB-D SLAM Systems, Proc. of the International Conference on Intelligent Robot Systems (IROS) 2012

21. Reddy, Pavan & Gujral, Aditya. (2025). Improving Neural Network Efficiency Using Piecewise Linear Approximation of Activation Functions. The International FLAIRS Conference Proceedings. 38. 10.32473/flairs.38.1.139005.

22. Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, Stephen Marshall // Activation Functions: Comparison of trends in Practice and Research for Deep Learning 2018, https://arxiv.org/abs/1811.03378

23. CUDA C++ Best Practices Guide // https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/ (дата обращения 26.05.2026)

24. DROID-SLAM Github // https://github.com/princeton-vl/DROID-SLAM (дата обращения 24.05.2026)

25. Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, Sebastian Scherer // TartanAir: A Dataset to Push the Limits of Visual SLAM, 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020

26. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern

Recognition, Providence, RI, USA, 2012, pp. 3354-3361, DOI: 10.1109/CVPR.2012.6248074.

27. W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", The International Journal of Robotics Research (IJRR), 2016.

References

1. Doroshev A.S., Shelomanov D.A. Methodology for selecting hyperparameters of a neural network model in optical navigation tasks. Trudy MAI. 2025. No. 142. (In Russ.). URL: https://trudymai.ru/eng/published.php?ID=185106

2. The $7 trillion data center build-out: How industrials can capture their share // https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-7-trillion-dollar-data-center-build-out-how-industrials-can-capture-their-share (дата обращения 24.05.2026)

3. Global RAM shortage appears set to continue through 2027 // https://www.computerworld.com/article/4161043/the-memory-shortage-appears-set-to-continue-through-2027.html (дата обращения 24.05.2026)

4. Data centers will consume 70 percent of memory chips made in 2026 - supply shortfall will cause the chip shortage to spread to other segments // https://www.tomshardware.com/pc-components/ram/data-centers-will-consume-70-percent-of-memory-chips-made-in-2026-supply-shortfall-will-cause-the-chip-shortage-to-spread-to-other-segments (дата обращения 30.05.2026)

5. Korytkin P.G. Applying deep learning networks in visual odometry problems // Trudy MAI. 2025. No. 145. (In Russ.). URL: https://trudymai.ru/published.php?ID=186898

6. Sen Wang, Ronald Clark, Hongkai Wen, Niki Trigoni // DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 2043–50. https://doi.org/10.1109/icra.2017.7989236

7. Olivier Brochu Dufour and Abolfazl Mohebbi and Sofiane Achiche, An Attention-Based Deep Learning Architecture for Real-Time Monocular Visual Odometry: Applications to GPS-free Drone Navigation, 2024 https://arxiv.org/abs/2404.17745

8. Teed, Zachary and Deng, Jia, “DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras”, Advances in neural information processing systems, 2021.

9. Tabassum, T.E.; Xu, Z.; Petrunin, I.; Rana, Z.A. Integrating GRU with a Kalman Filter to Enhance Visual Inertial Odometry Performance in Complex Environments. Aerospace 2023, 10, 923. https://doi.org/10.3390/aerospace10110923

10. J. C. Heck and F. M. Salem, "Simplified minimal gated unit variations for recurrent neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1593-1596, doi: 10.1109/MWSCAS.2017.8053242.

11. Moyang Li, Zihan Zhu, Marc Pollefeys, Daniel Barath // DROID-SLAM in the Wild. 2026 https://arxiv.org/abs/2603.19076

12. Y. Huang, L. Ji, J. Sun, X. Jiang and H. Liu, "CUEMP: Correspondence Uncertainty Estimation With Motion Priors for Dense Visual Odometry," in IEEE Robotics and Automation Letters, vol. 11, no. 4, pp. 4497-4504, April 2026, doi: 10.1109/LRA.2026.3666386.

13. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, Max Tegmark // KAN: Kolmogorov-Arnold Networks, 2024 https://arxiv.org/abs/2404.19756

14. Yesmine Abdennadher, Eleonora Cicciarella, Michele Rossi // Convolutional Spiking-based GRU Cell for Spatio-temporal Data, 2025 IEEE 35th International Workshop on Machine Learning for Signal Processing (MLSP), 2025 http://dx.doi.org/10.1109/MLSP62443.2025.11204318

15. Albukhari, I.; El-Sayed, A.; Alshibli, M. MINI-DROID-SLAM: Improving Monocular Visual SLAM Using MINI-GRU RNN Network. Sensors 2025, 25, 5448. https://doi.org/10.3390/s25175448

16. Mateos-Ramirez, P.; Gomez-Avila, J.; Villaseñor, C.; Arana-Daniel, N. Visual Odometry in GPS-Denied Zones for Fixed-Wing Unmanned Aerial Vehicle with Reduced Accumulative Error Based on Satellite Imagery. Appl. Sci. 2024, 14, 7420. https://doi.org/10.3390/app14167420

17. Burri, Michael and Nikolic, Janosch and Gohl, Pascal and Schneider, Thomas and Rehder, Joern and Omari, Sammy and Achtelik, Markus W and Siegwart, Roland //

The EuRoC micro aerial vehicle datasets, 2016 https://doi.org/10.1177/0278364915620033

18. Grupp, Michael. evo: Python package for the evaluation of odometry and SLAM., 2017 https://github.com/MichaelGrupp/evo

19. J. Sturm and N. Engelhard and F. Endres and W. Burgard and D. Cremers // A Benchmark for the Evaluation of RGB-D SLAM Systems, Proc. of the International Conference on Intelligent Robot Systems (IROS) 2012

20. Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, Stephen Marshall // Activation Functions: Comparison of trends in Practice and Research for Deep Learning 2018, https://arxiv.org/abs/1811.03378

21. Reddy, Pavan & Gujral, Aditya. (2025). Improving Neural Network Efficiency Using Piecewise Linear Approximation of Activation Functions. The International FLAIRS Conference Proceedings. 38. 10.32473/flairs.38.1.139005.

22. Roberto Pellerito, Marco Cannici, Daniel Gehrig, Joris Belhadj, Olivier Dubois-Matra, Massimo Casasco, Davide Scaramuzza, “End-to-end Learned Visual Odometry with Events and Frames”, 2024, https://arxiv.org/abs/2309.09947

23. CUDA C++ Best Practices Guide // https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/ (дата обращения 26.05.2026)

24. DROID-SLAM Github // https://github.com/princeton-vl/DROID-SLAM (дата обращения 24.05.2026)

25. Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, Sebastian Scherer // TartanAir: A Dataset to Push the Limits of Visual SLAM, 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020

26. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3354-3361, DOI: 10.1109/CVPR.2012.6248074.

27. W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", *The International Journal of Robotics Research (IJRR)*, 2016.

Информация об авторах

Павел Георгиевич Корыткин, аспирант, Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В. Ломоносова», г. Москва, Россия; ORCID: https://orcid.org/0009-0000-1761-463X; e-mail: korytkinpg@my.msu.ru

Information about the authors

Pavel G. Korytkin, Postgraduate Student; Federal State Budget Educational Institution of Higher Education M.V. Lomonosov Moscow State University, Moscow, Russia;ORCID: https://orcid.org/0009-0000-1761-463X;
e-mail: korytkinpg@my.msu.ru

Получено 11 июня 2026 ● Принято к публикации 04 октября 2026 ● Опубликовано 28 октября 2026
Received 11 June 2026 ● Accepted 04 October 2026 ● Published 30 October 2026