DOI: 10.15514/ISPRAS-2026-38(4)-2

Послойная обобщенная гладкость: единый подход для адаптивных LMO-оптимизаторов

А.А. Рябинин <artem.riabinin152@gmail.com>

Лаборатория фундаментальных исследований искусственного интеллекта,
Россия, 117218, г. Москва, ул. Большая Черемушкинская, д. 20.

Аннотация. Недавно появившиеся алгоритмы, основанные на оракуле линейной минимизации (Linear Minimization Oracle, LMO), такие как Muon и Scion, становятся конкурентоспособной заменой оптимизатору Adam. Они обеспечивают более эффективное использование памяти, лучшую переносимость гиперпараметров и превосходное эмпирическое качество в крупномасштабных задачах, например при обучении больших языковых моделей (LLM). Тем не менее между их практическим успехом и теоретическим пониманием сохраняется существенный разрыв: предшествующий анализ игнорирует послойное применение этих оптимизаторов и опирается на нереалистичные предположения о гладкости, приводящие к непрактично малым шагам. Чтобы устранить эти проблемы, мы предлагаем обобщённый послойный LMO-подход вместе с уточнённой моделью обобщённой гладкости. Такой подход точно учитывает послойную геометрию нейронных сетей и даёт гарантии сходимости с высокой предсказательной силой. Кроме того, в отличие от предыдущих результатов, предсказанные теоретические шаги близко соответствуют тонко настроенным значениям. Эксперименты с NanoGPT и CNN подтверждают, что наше предположение выполняется вдоль траектории оптимизации, что в итоге устраняет разрыв между теорией и практикой.

Ключевые слова: оптимизация; оракул линейной минимизации; послойная гладкость; большие языковые модели.

Для цитирования: Рябинин А.А. Послойная обобщенная гладкость: единый подход для адаптивных LMO-оптимизаторов. Труды ИСП РАН, 2026, том 38, вып. 4, часть 1, стр. 25–48. DOI: 10.15514/ISPRAS-2026-38(4)-2.

Layer-Wise Generalized Smoothness: A Unified Framework for Adaptive LMO Optimizers

A.A. Riabinin <artem.riabinin152@gmail.com>

Basic Research of Artificial Intelligence Laboratory (BRAIn Lab),
20 Bolshaya Cheremushkinskaya St., Moscow, 117218, Russia.

Abstract. Recent algorithms based on the Linear Minimization Oracle (LMO) framework, such as Muon and Scion, are emerging as viable replacements for the Adam optimizer. They offer improved memory efficiency, better hyperparameter transferability, and superior empirical performance in large-scale tasks like large language model (LLM) training. However, a significant gap remains between their practical success and theoretical understanding. To address these issues, we propose a generalized layer-wise LMO framework alongside a refined generalized smoothness model. This approach accurately captures the layer-wise geometry of neural networks, yielding convergence guarantees with strong predictive power. Experiments with NanoGPT and CNN confirm that our assumption holds along the optimization trajectory, ultimately closing the gap between theory and practice.

Keywords: adaptive optimization; linear minimization oracle; layer-wise smoothness; large language models.

For citation: Riabinin A.A. Layer-Wise Generalized Smoothness: A Unified Framework for Adaptive LMO Optimizers. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 4, part 1, pp. 25-48 (in Russian). DOI: 10.15514/ISPRAS-2026-38(4)-2.

1. Введение

Выдающиеся достижения современных архитектур глубокого обучения во многом зависят от применяемых методов оптимизации. По мере роста нейронных сетей как по глубине, так и по объёму требуемых данных оптимизация превратилась в критическое узкое место, регулярно потребляющее тысячи GPU-часов; особенно отчётливо это проявляется при обучении больших языковых моделей (large language models, LLM). На протяжении более десяти лет в этой области доминировали метод Adam [1] и его вариант с разделённым затуханием весов AdamW [2]. Несмотря на широкое распространение и общую надёжность, эти методы адаптивной оценки моментов обладают заметными недостатками. Главными среди них являются сильная зависимость от исчерпывающего подбора гиперпараметров, высокая чувствительность к расписаниям скорости обучения и возможное ухудшение обобщающей способности при недостаточно тщательной настройке [3, 4].

В последнее время начался многообещающий сдвиг парадигмы с появлением новых оптимизаторов, таких как Muon [5] и Scion [6]. Отказываясь от адаптивной оценки моментов в пользу учитывающего геометрию подхода в духе Франка – Вульфа [7, 8], эти алгоритмы представляют собой убедительную альтернативу. Эмпирически они демонстрируют большую устойчивость к выбору гиперпараметров и снижают потребность в исчерпывающей настройке, превосходя при этом AdamW в крупномасштабных сценариях обучения LLM [9, 6].

Тем не менее теоретическое обоснование этих методов на основе оракула линейной минимизации (linear minimization oracle, LMO) не успевает за их эмпирическим успехом. Существующие гарантии сходимости в реалистичных невыпуклых режимах по-прежнему оставляют значительный разрыв между теорией и эмпирическими результатами. Как показано в разделе 3, существующие теоретические подходы в основном анализируют упрощённые абстракции, не отражающие реальные реализации. Мы выделяем два главных расхождения: игнорирование послойной структуры (раздел 3.1) и неудачный выбор шагов, обусловленный неточной моделью гладкости (раздел 3.2). В данной работе мы устраняем этот критический разрыв между теорией и практикой, формулируя обобщённый послойный LMO-подход в сочетании с уточнённой моделью гладкости.

Наша цель – решить общую задачу стохастической оптимизации, определяемую как:

$$\min_{X\in\mathcal{S}}\{f(X) := \mathbb{E}_{\xi\sim\mathcal{D}}[f_\xi(X)]\}, \quad (1)$$

где $\mathcal{S}$ – конечномерное векторное пространство, а $f_\xi: \mathcal{S} \mapsto \mathbb{R}$ – потенциально невыпуклые и негладкие, но непрерывно дифференцируемые функции. Здесь $f_\xi(X)$ обозначает потерю, вычисленную на выборке данных ξ, взятом из распределения вероятностей $\mathcal{D}$, для модели с параметрами X. Чтобы задача была корректно поставлена, предполагаем, что глобальный инфимум ограничен снизу, то есть $f^{\inf} := \inf_{X\in\mathcal{S}} f(X) > -\infty$. С учётом иерархической природы глубоких нейронных сетей глобальный вектор параметров $X \in \mathcal{S}$ образуется конкатенацией весовых матриц $X_i \in \mathcal{S}_i := \mathbb{R}^{m_i\times n_i}$, соответствующих каждому обучаемому слою $i = 1, \dots, p$. Соответственно, мы обозначаем $X = [X_1, \dots, X_p]$. Это означает, что полное пространство параметров $\mathcal{S}$ является d-мерным тензорным произведением пространств отдельных слоёв:

$$\mathcal{S} := \mathcal{S}_1 \otimes \mathcal{S}_2 \otimes \dots \otimes \mathcal{S}_p,$$

где полная размерность равна $d := \sum_{i=1}^{p} m_i n_i$. Кроме того, мы наделяем каждое подпространство слоя $\mathcal{S}_i$ следовым скалярным произведением $\langle X_i, Y_i\rangle_{(i)} := \mathrm{tr}(X_i^\top Y_i)$ и произвольной нормой $\|\cdot\|_{(i)}$, которая не обязана порождаться этим скалярным произведением.

2. Обзор литературы

Классическое предположение об L-гладкости, при котором градиент приобретает свойство липшицевости с глобальной константой L, часто не способно точно отразить сложную геометрию ландшафтов потерь в глубоком обучении [10-11]. Чтобы устранить это, в [11] ввели условие (L^0, L^1)-гладкости, эмпирически наблюдая в экспериментах с языковыми моделями, что оценка вида $\| \nabla^2 f(x) \| \le L_0 + L_1 \| \nabla f(x) \|$ лучше описывает поведение нормы гессиана. Последующие работы анализировали стандартные алгоритмы оптимизации в этом подходе обобщённой гладкости. Например, в [12] и [13] приведены анализы сходимости для градиентного метода. В [14] проанализирован нормализованный стохастический градиентный спуск (Stochastic gradient descent, SGD) с моментом в параметрически-агностической постановке при (L^0, L^1)-гладкости. В [15] предложена неевклидова обобщённая гладкость и установлены скорости сходимости для методов зеркального спуска. Наша работа расширяет это направление, встраивая (L^0, L^1)-гладкость в послойный контекст с произвольными нормами – подход, особенно хорошо подходящий для изучаемых нами LMO-оптимизаторов.

Признавая неоднородную природу параметров в больших моделях, исследователи изучали условия анизотропной гладкости, где константы гладкости могут различаться по разным размерностям или блокам параметров. К ранним работам в этом направлении относится покоординатная липшицевость для методов покоординатного спуска [16-17]. Позднее в [18] проанализировали алгоритм signSGD при более слабом понятии покоординатной гладкости. В [10] развили это, проанализировав обобщённый signSGD при обобщённом покоординатном условии гладкости и подчеркнув, что разные группы параметров могут обладать существенно разной геометрией. В [19] сосредоточились на анализе алгоритма адаптивного градиента (Adaptive Gradient, Adagrad) при покоординатной гладкости и установили нижние оценки для SGD, подчёркивая преимущества адаптивности. В [20] предложили «анизотропную (L^0, L^1)-гладкость» (векторную версию (L_0, L_1)-гладкости, применяемую покоординатно) и продемонстрировали доказуемые преимущества Adagrad над SGD в этой постановке. В работе [21] также использовали понятия анизотропной гладкости в анализе сходимости алгоритма Adam. Наша работа вносит вклад, определяя и анализируя послойную (L^0, L^1)-гладкость, которая сочетает преимущества модели обобщённой гладкости со

структурированной, анизотропной перспективой, подстроенной под послойно-блочную архитектуру нейронных сетей и совместимой с произвольными послойными нормами. Этот подход существен для понимания LMO-методов вроде Muon и Scion.

Оптимизаторы Muon [5] и Scion [6] представляют недавно представленный класс методов, показавших сильное эмпирическое качество в глубоком обучении. Muon изначально был представлен как эффективный эмпирический метод, правило обновления которого для скрытых слоёв вдохновлено идеями из [22]. Впоследствии в [6] (авторы Scion) явно связали эти типы обновлений с подходом Франка–Вульфа (FW) [7, 23], предложив использовать послойные нормы в правиле обновления на основе LMO. Эти методы выполняют обновления, решая для каждого слоя задачу линейной минимизации над шаром по норме с центром в текущей итерации. Предшествующие теоретические анализы этих оптимизаторов [24, 25, 6] опирались на стандартную L-гладкость и анализировали упрощённое глобальное обновление. Наша работа даёт первые гарантии сходимости для этих методов при более реалистичной послойной (L_0, L_1)-гладкости, непосредственно учитывая их практическую послойную природу и используя геометрические соображения, предоставляемые LMO над общими нормами.

3. Теория и практика в LMO-оптимизации

В данной работе мы сосредоточимся на алгоритмическом подходе, который итеративно вызывает оракулы линейной минимизации (LMO) по всем слоям сети. Эта процедура, формализованная нами как послойный LMO-подход в Алгоритме 1, вычисляет обновления параметров для каждого независимого слоя i через следующую задачу минимизации с ограничениями:

$$X_i^{k+1} = \mathrm{LMO}_{\mathcal{B}_i^k}(M_i^k) := \operatorname*{argmin}_{X_i\in\mathcal{B}_i^k} \langle M_i^k, X_i\rangle_{(i)}, \text{ где } \mathcal{B}_i^k := \{X_i \in \mathcal{S}_i : \| X_i - X_i^k \|_{(i)} \le t_i^k\}, \quad (2)$$

где $t_i^k > 0$ обозначает адаптивно определяемый шаг, радиус или скорость обучения. В этом контексте радиусы, задающие допустимые шары по норме, определяют величину обновления параметров; поэтому термины «радиус», «шаг» и «скорость обучения» используются в работе как взаимозаменяемые. Важно подчеркнуть, что переменная момента $M^k = [M_1^k, \dots, M_p^k] \in \mathcal{S}$ агрегирует историческую траекторию стохастических градиентов $\nabla f_{\xi^k}(X^k) = [\nabla_1 f_{\xi^k}(X^k), \dots, \nabla_p f_{\xi^k}(X^k)] \in \mathcal{S}$ (как описано на шаге 5 Алгоритма 1).

Наш обобщённый подход включает в себя несколько существующих методов, в частности охватывая оптимизаторы Muon и Scion как специальные случаи, получаемые при определённом выборе норм (см. раздел 4.1). Помимо демонстрации превосходного эмпирического качества по сравнению с AdamW на массивных бенчмарках, эти LMO-оптимизаторы обладают рядом крайне желательных свойств, включая улучшенное использование памяти, повышенную устойчивость к конфигурациям гиперпараметров и надёжную переносимость этих конфигураций между моделями разной ёмкости [6, 26]. Более того, в отличие от исторического пути Adam, эти алгоритмы почти сразу после появления сопровождались теоретическими гарантиями сходимости, как правило установленными при стандартных условиях липшицевой гладкости и ограниченной дисперсии стохастического градиента [24-25, 6].

Хотя наш формализованный метод (см. Алгоритм 1) точно отражает весьма эффективные подходы, применяемые на практике [27-28], существующий теоретический анализ [24-25, 6] не моделирует эти реальные реализации адекватно. В частности, предшествующая литература расходится с реальными реализациями в двух фундаментальных аспектах, что серьёзно ограничивает её способность математически объяснить эмпирический успех алгоритмов. Далее мы подробно изложим эти расхождения.

Вход: Начальные параметры модели $X^0 = [X_1^0, \dots, X_p^0] \in \mathcal{S}$, момент $M^0 = [M_1^0, \dots, M_p^0] \in \mathcal{S}$, коэффициенты затухания момента $\beta^k \in [0,1)$ для всех итераций $k \geq 0$

1: для $k = 0,1,2, \dots, K-1$ выполнять

2: генерируем $\xi^k \sim \mathcal{D}$

3: для $i = 1,2, \dots, p$ выполнять

4: вычисляем стохастический градиент $\nabla_i f_{\xi^k}(X^k)$ для слоя i

5: обновляем момент $M_i^k = \beta^k M_i^{k-1} + (1-\beta^k)\nabla_i f_{\xi^k}(X^k)$ для слоя i

6: выбираем адаптивный шаг/радиус $t_i^k > 0$ для слоя i

7: обновляем параметры слоя i через LMO по $\mathcal{B}_i^k := \{X_i \in \mathcal{S}_i : \| X_i - X_i^k \|_{(i)} \leq t_i^k\}$:

$$X_i^{k+1} = \mathrm{LMO}_{\mathcal{B}_i^k}(M_i^k) := \operatorname*{argmin}_{X_i \in \mathcal{B}_i^k} \langle M_i^k, X_i \rangle_{(i)} \quad (2)$$

8: конец цикла

9: обновляем полный вектор параметров $X^{k+1} = [X_1^{k+1}, \dots, X_p^{k+1}]$

10: конец цикла

Алгоритм 1. Стохастический адаптивный послойный LMO-оптимизатор с моментом.
Algorithm 1. Stochastic adaptive layer-wise LMO optimizer with momentum.

3.1. Послойная структура

Начнём с обзора теоретических оснований, заложенных в предшествующих исследованиях. Muon изначально разрабатывался специально для скрытых слоёв, полагаясь на стандартные оптимизаторы вроде Adam или AdamW для входного и выходного слоёв. Его первоначальное представление в работе [5] было чисто эмпирическим и не содержало формального теоретического анализа. Первые гарантии сходимости, полученные в [25], рассматривали гладкий невыпуклый случай, но ограничивались задачей (1) с $p = 1$, фактически сводя анализ к однослойному сценарию. Впоследствии оптимизатор Scion [6] – возникающий как специальный случай в нашем подходе – улучшил Muon, распространив правило обновления на основе минимизирующей процедуры LMO на все слои, что дало превосходное эмпирическое качество. (В работе [6] вводятся два варианта оптимизатора Scion: один для оптимизации с ограничениями, называемый просто «Scion», и другой для задач без ограничений, называемый «unconstrained Scion». В данной работе под «Scion» понимается любой из вариантов, а «unScion» используется при ссылке на вариант без ограничений.) И [6], и [24] строят свой анализ сходимости на варианте глобального правила обновления:

$$M^k = \beta^k M^{k-1} + (1-\beta^k)\nabla f_{\xi^k}(X^k), \qquad X^{k+1} = \mathrm{LMO}_{\mathcal{B}^k}(M^k), \quad (3)$$

где $\beta^k \in [0,1)$ обозначает параметр момента, $\nabla f_{\xi^k}(X^k)$ – стохастический градиент, вычисленный на итерации k, а $\mathcal{B}^k := \{X \in \mathcal{S} : \| X - X^k \| \leq t^k\}$ представляет глобальный шар по норме с центром в X^k и единым радиусом $t^k > 0$.

Эта постановка близко напоминает структуру нашего подхода, но не совпадает с ней. Действительно, наш подход обновляет параметры послойно, а не совместно по всему вектору X. Это различие критично, поскольку для практических, чрезвычайно высокоразмерных моделей вычисление единого глобального LMO для всего вектора параметров вычислительно неосуществимо. Разложение оптимизации на локализованные послойные LMO – именно то, что восстанавливает вычислительную осуществимость.

Руководствуясь этим теоретическим разрывом, мы строим наш анализ в пространстве произведения матриц $\mathcal{S}$, чтобы явно отразить послойную структуру. Такой подход позволяет

строго рассмотреть фактические послойные обновления, определённые в (2), допуская локализованные геометрические предположения и послойные конфигурации гиперпараметров.

3.2. Теория с предсказательной силой

Существующие гарантии сходимости для Алгоритма 1 страдают от значительных аналитических ограничений, проявляющихся в виде практических недостатков. В частности, предшествующие теоретические оценки LMO-методов вроде Muon и Scion опираются на стандартное условие L-гладкости. Это накладывает единую константу гладкости на всю сеть, что проблематично, поскольку отдельные слои обладают различными геометрическими свойствами и требуют специализированного подхода.

Расхождение не ограничивается единым подходом к слоям. Целевые функции современного глубокого обучения широко признаны негладкими [10-11]. Из-за этого теоретического несоответствия предшествующий анализ предписывает непрактично малые единые шаги (как показано в табл. 1). Эти теоретические скорости обучения полностью оторваны от динамически настраиваемых значений, обеспечивающих передовое эмпирическое качество. В результате существующие теории практически не объясняют практический успех этих методов. Вместо того чтобы давать практические рекомендации по выбору гиперпараметров, они оставляют практиков зависимыми от ресурсоёмкой ручной настройки.

Табл. 1. Сравнение гарантий сходимости для нашего подхода (Алгоритм 1 и 2) для достижения $\min_{k=0,\dots,K-1} \sum_{i=1}^p \|\nabla_i f(X^k)\|_{(i)\star}$; нотация $O(\cdot)$ скрывает логарифмические множители.
Table 1. Comparison of convergence guarantees for our framework (Algorithms 1 and 2) for achieving $\min_{k=0,\dots,K-1} \sum_{i=1}^p \|\nabla_i f(X^k)\|_{(i)\star}$; the $O(\cdot)$ notation hides logarithmic factors.

Результат	**Стохаст.?**	(L^0, L^1)	**Скорость**	**Шаги** t_i^k	$1-\beta^k$
[24, Теорема 1]	Нет	Нет	$\mathcal{O}(1/K^{1/2})$	const $\propto 1/K^{1/2}$	–
[24, Теорема 2], [25, Теорема 2.1]	Да	Нет	$\mathcal{O}(1/K^{1/4})$	const $\propto 1/K^{3/4}$	const $\propto 1/K^{1/2}$
[6, Лемма 5.4]	Да	Нет	$\mathcal{O}(1/K^{1/4})$	const $\propto 1/K^{3/4}$	$\propto 1/k^{1/2}$
Теорема 1	Нет	Да	$\mathcal{O}(1/K^{1/2})$	Адаптивные	–
Теорема 3	Да	Да	$\mathcal{O}(1/K^{1/4})$	$\propto 1/k^{3/4}$	$\propto 1/k^{1/2}$

Как показывают наши результаты в Теореме 1 (см. раздел 4.2), эту теоретическую стагнацию можно преодолеть. Чтобы точнее смоделировать ландшафт оптимизации глубоких нейронных сетей, мы вводим уточнённое условие регулярности: послойную (L^0, L^1)-гладкость. Это условие расширяет подход обобщённой гладкости, предложенный в [11], применяя его явно на уровне слоёв. (Хотя мы формулируем Предположение 1 в глобальной форме для наглядности обозначений, наши доказательства строго требуют выполнения предположения лишь локально. В частности, предположение ограничивает изменение градиента между последовательными итерациями X^k и X^{k+1}. Поскольку наши шаги ограниченны, расстояние между итерациями остаётся контролируемым, что обеспечивает достаточность локального варианта этого условия для всех теоретических гарантий.)

Предположение 1 (Послойная (L^0, L^1)-гладкость). Функция $f: \mathcal{S} \mapsto \mathbb{R}$ является послойно (L^0, L^1)-гладкой с константами $L^0 := (L_1^0, \dots, L_p^0) \in \mathbb{R}_+^p$ и $L^1 := (L_1^1, \dots, L_p^1) \in \mathbb{R}_+^p$. То есть неравенство

$$\| \nabla_i f(X) - \nabla_i f(Y) \|_{(i)\star} \leq \left(L_i^0 + L_i^1 \| \nabla_i f(X) \|_{(i)\star}\right) \| X_i - Y_i \|_{(i)} \quad (4)$$

выполняется для всех $i = 1, \dots, p$ и всех $X = [X_1, \dots, X_p] \in \mathcal{S}$, $Y = [Y_1, \dots, Y_p] \in \mathcal{S}$, где $\|\cdot\|_{(i)\star}$ – двойственная норма к $\|\cdot\|_{(i)}$ (то есть $\| X_i \|_{(i)\star} := \sup_{\|Z_i\|_{(i)} \le 1} \langle X_i, Z_i \rangle_{(i)}$ для любого $X_i \in \mathcal{S}_i$).

Предположение 1 является обобщением анизотропной «векторной» (L^0, L^1)-гладкости, представленной в [20], адаптированной для произвольных норм, которая, в свою очередь, обобщает модель (L^0, L^1)-гладкости из [11]. Следовательно, наш аналитический подход выходит за рамки предшествующих работ, строго ограниченных классической L-гладкостью. Что ещё важнее, это обобщение не является чисто теоретическим; мы утверждаем, что оно точно отражает базовую геометрию современных нейронных сетей. Это утверждение подкрепляется двумя основными причинами.

Во-первых, в отличие от классической L-гладкости, наша формулировка тесно согласуется с эмпирическими данными. На рис. 1 мы демонстрируем эмпирическую проверку Предположения 1, анализируя обучение NanoGPT на наборе данных FineWeb. Мы строим оценку гладкости вдоль траектории $\hat{L}_i[k]$ (определённую в (14)) вместе с теоретической аппроксимацией $\hat{L}_i^{\text{approx}}[k] := L_i^0 + L_i^1 \| \nabla_i f_{\hat{\xi}^{k+1}}(X^{k+1}) \|_{(i)\star}$, используя послойные параметры L_i^0 и L_i^1, полученные из обучающих данных. Графики иллюстрируют эти величины для слоя эмбеддингов и отдельных трансформерных блоков. Тесное согласие между $\hat{L}_i[k]$ и $\hat{L}_i^{\text{approx}}[k]$ указывает на то, что Предположение 1 приблизительно выполняется на всей траектории оптимизации. Как подробно описано в разделе 5, мы дополнительно подтверждаем это согласие для всей архитектуры модели как для NanoGPT, так и для CNN, обученной на CIFAR-10. Неизменно мы наблюдаем, что $L_i^0 \approx 0$ для всех слоёв, что дополнительно выявляет неадекватность классических предположений о гладкости. Кроме того, рис. 1 показывает, что гладкость вдоль траектории значительно колеблется между разными блоками и слоями, подкрепляя необходимость послойного подхода. В совокупности эти результаты указывают на то, что послойная (L^0, L^1)-гладкость даёт гораздо более точное представление о ландшафте потерь глубокого обучения.

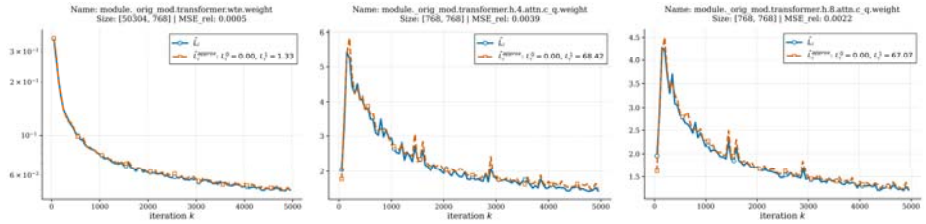

Рис. 1. Обучение NanoGPT на FineWeb подтверждает нашу модель послойной (L^0, L^1)-гладкости. (a) Матрица эмбеддингов токенов из первого/последнего слоя; (b) матрица запросов self-attention из 4-го трансформерного блока; (c) матрица запросов self-attention из 8-го трансформерного блока.

Fig. 1. Training NanoGPT on FineWeb confirms our layer-wise (L^0, L^1)-smoothness model. (a) Token embedding matrix of the first/last layer; (b) self-attention query matrix of the 4th transformer block; (c) self-attention query matrix of the 8th transformer block.

Во-вторых, Предположение 1 не только описывает геометрию модели; она активно направляет формулировку адаптивных, высокоэффективных шагов. Через Теорему 1 мы выводим скорости обучения, подстроенные под локальную геометрию каждой отдельной группы параметров и непосредственно опирающиеся на наше условие послойной гладкости. Как показано в разделе 5.1, эти теоретически предписанные шаги точно отражают относительные величины оптимально настроенных скоростей обучения, приведённых в [6]. Это согласие служит убедительным подтверждением нашей теоретической модели и явно подчёркивает практическую ценность послойного проектирования оптимизации.

4. Основная теория и результаты

Чтобы лучше понять структуру обновлений, начнём с детерминированной формулировки нашего метода, изложенной в Алгоритме 2. На каждой итерации этот подход независимо минимизирует линейное приближение f вокруг каждой группы параметров X_i^k внутри шара радиуса $t_i^k > 0$, что естественно допускает послойное проектирование алгоритма.

Вход: Начальные параметры модели $X^0 = [X_1^0, \dots, X_p^0] \in \mathcal{S}$
1: для $k = 0,1, \dots, K-1$ выполнять
  2: для $i = 1,2, \dots, p$ выполнять
    3: выбираем адаптивный шаг/радиус $t_i^k > 0$ для слоя i
    4: обновляем параметры слоя i через LMO по $\mathcal{B}_i^k := \{X_i \in \mathcal{S}_i : \| X_i - X_i^k \|_{(i)} \le t_i^k\}$:
      $$X_i^{k+1} = \text{LMO}_{\mathcal{B}_i^k}(\nabla_i f(X^k)) := \operatorname*{argmin}_{X_i \in \mathcal{B}_i^k} \langle \nabla_i f(X^k), X_i \rangle_{(i)} \quad (5)$$
    5: конец цикла
  6: обновляем полный вектор: $X^{k+1} = [X_1^{k+1}, \dots, X_p^{k+1}]$
7: конец цикла

Алгоритм 2. Детерминированный адаптивный послойный LMO-оптимизатор.
Algorithm 2. Deterministic adaptive layer-wise LMO-optimizer.

4.1. Примеры оптимизаторов, удовлетворяющих нашему подходу

Детерминированная версия нашего подхода описывает общий класс методов, параметризованный выбором норм $\|\cdot\|_{(i)}$ в LMO. Чтобы проиллюстрировать гибкость этого подхода, выделим несколько примечательных частных случаев. Сначала заметим, что правило обновления (5) можно записать как

$$X_i^{k+1} = X_i^k + t_i^k \text{LMO}_{\{X_i \in \mathcal{S}_i : \|X_i\|_{(i)} \le 1\}}(\nabla_i f(X^k)) = X_i^k + t_i^k \operatorname*{argmin}_{\|X_i\|_{(i)} \le 1} \langle \nabla_i f(X^k), X_i \rangle_{(i)}. \quad (6)$$

Для любого $X_i \in \mathcal{S}_i = \mathbb{R}^{m_i \times n_i}$ определим $\| X_i \|_{\alpha \to \beta} := \sup_{\|z\|_\alpha = 1} \| X_i z \|_\beta$, где $\|\cdot\|_\alpha$ и $\|\cdot\|_\beta$ – некоторые (возможно, неевклидовы) нормы на $\mathbb{R}^{n_i}$ и $\mathbb{R}^{m_i}$ соответственно. Заметим, что (6) естественно восстанавливает несколько известных обновлений при определённом выборе норм слоёв. Теперь подробнее обсудим наиболее примечательные частные случаи.

Послойный нормализованный GD [29]. Пусть $\|\cdot\|_{(i)} = \|\cdot\|_{2 \to 2}$ для каждой группы параметров, и предположим, что $n_i = 1$ для всех $i = 1, \dots, p$. В этом случае спектральная норма сводится к стандартной евклидовой норме $\|\cdot\|_2$, что даёт правило обновления

$$X_i^{k+1} = X_i^k - t_i^k \frac{\nabla_i f(X^k)}{\| \nabla_i f(X^k) \|_2}, \quad i = 1, \dots, p,$$

которое соответствует послойному нормализованному GD. При подходящем выборе t_i^k (см. Теорему 1) метод также может восстанавливать градиентный метод для (L^0, L^1)-гладких функций [13].

Послойный signGD [30]. Предположим, что $\|\cdot\|_{(i)} = \|\cdot\|_{1 \to \infty}$ для каждой группы параметров, при $n_i = 1$ для всех $i = 1, \dots, p$. Тогда $\|\cdot\|_{1 \to \infty}$ сводится к $\|\cdot\|_\infty$, и обновление принимает вид

$$X_i^{k+1} = X_i^k - t_i^k \operatorname{sign}(\nabla_i f(X^k)), \quad i = 1, \dots, p,$$

где функция знака применяется поэлементно. Это эквивалентно послойному signGD.

Muon [5]. Здесь для всех групп параметров используется спектральная норма $\|\cdot\|_{2 \to 2}$ без ограничений на n_i. В этом случае можно показать, что (5) эквивалентно

$$X_i^{k+1} = X_i^k - t_i^k U_i^k (V_i^k)^\top, \quad i = 1, \dots, p, \quad (7)$$

где $\nabla_i f(X^k) = U_i^k \Sigma_i^k \left(V_i^k\right)^\top$ – сингулярное разложение [22]. Это в точности послойная детерминированная версия оптимизатора Muon. В практическом обучении LLM более общий вариант (7), включающий стохастичность и момент, применяется к промежуточным слоям, тогда как входной и выходной слои оптимизируются другими методами.

Unconstrained Scion [6]. Мы также можем восстановить два варианта unScion, введённых в [6]: один для обучения LLM на предсказании следующего токена, другой для обучения CNN для классификации изображений.

Обучение LLM. Определим нормы $\|\cdot\|_{(i)}$ следующим образом: для $i = 1, \dots, p-1$, соответствующих весовым матрицам трансформерных блоков, положим $\|\cdot\|_{(i)} = \sqrt{n_i/m_i}\,\|\cdot\|_{2\to 2}$, а для последней группы X_p, представляющей слои эмбеддинга и вывода (которые совпадают в рассматриваемом здесь режиме разделения весов), положим $\|\cdot\|_{(p)} = n_p\,\|\cdot\|_{1\to\infty}$. В этом случае (5) принимает вид

$$\begin{aligned} X_i^{k+1} &= X_i^k - t_i^k \sqrt{\frac{m_i}{n_i}} U_i^k \left(V_i^k\right)^\top, \quad i = 1, \dots, p-1, \\ X_p^{k+1} &= X_p^k - \frac{t_p^k}{n_p} \operatorname{sign}\left(\nabla_p f(X^k)\right), \end{aligned} \tag{8}$$

где $\nabla_i f(X^k) = U_i^k \Sigma_i^k \left(V_i^k\right)^\top$ – сингулярное разложение. Это эквивалентно детерминированному послойному оптимизатору unScion без момента. Более общий вариант, включающий стохастичность и момент и применяемый ко всем слоям, как показано в [6], превосходит Muon в задачах обучения LLM.

Обучение CNN. Главное отличие в случае CNN – наличие не только 2D-весовых матриц, но и 1D-векторов смещений и 4D-параметров свёрточных ядер. Смещения – это 1D-тензоры формы $\mathbb{R}^{C_i^{out}}$, для которых мы используем масштабированные евклидовы нормы. Свёрточные параметры (conv) – это 4D-тензоры формы $\mathbb{R}^{C_i^{out} \times C_i^{in} \times k \times k}$, где C_i^{out} и C_i^{in} обозначают число выходных и входных каналов, а k – размер ядра. Для вычисления норм мы преобразуем каждый 4D-тензор в 2D-матрицу формы $\mathbb{R}^{C_i^{out} \times C_i^{in} k^2}$, а затем применяем масштабированную норму $\|\cdot\|_{2\to 2}$. Это даёт выбор норм $\|\cdot\|_{(i)} = \sqrt{1/C_i^{out}}\,\|\cdot\|_2$ для смещений, $\|\cdot\|_{(i)} = k^2 \sqrt{C_i^{in}/C_i^{out}}\,\|\cdot\|_{2\to 2}$ для conv и $\|\cdot\|_{(p)} = n_p\,\|\cdot\|_{1\to\infty}$ для последней группы X_p, связанной с весами классификационной головы. Тогда можно показать, что (5) эквивалентно

$$\begin{aligned} X_i^{k+1} &= X_i^k - t_i^k \sqrt{C_i^{out}} \frac{\nabla_i f(X^k)}{\|\nabla_i f(X^k)\|_2}, \text{(для смещений)}, \\ X_i^{k+1} &= X_i^k - t_i^k \frac{1}{k^2} \sqrt{\frac{C_i^{out}}{C_i^{in}}} U_i^k \left(V_i^k\right)^\top, \text{(для conv)}, \\ X_p^{k+1} &= X_p^k - \frac{t_p^k}{n_p} \operatorname{sign}\left(\nabla_p f(X^k)\right), \text{(для головы)} \end{aligned} \tag{9}$$

где $\nabla_i f(X^k) = U_i^k \Sigma_i^k \left(V_i^k\right)^\top$ – сингулярное разложение. Это соответствует детерминированному послойному оптимизатору unScion без момента.

4.2. Результаты о сходимости

Продемонстрировав гибкость подхода на конкретных примерах, сформулируем теперь общий результат о сходимости нашего детерминированного метода. Доказательства всех результатов вынесены в Приложение А.

Теорема 1. Пусть выполнено Предположение 1 и зафиксировано $\varepsilon > 0$. Пусть $X^0, \dots, X^{K-1}$ – итерации детерминированного LMO-оптимизатора (Алгоритм 2), запущенного с шагами $t_i^k = \frac{\|\nabla_i f(X^k)\|_{(i)\star}}{L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star}}$. Тогда:

1) Чтобы достичь точности $\min_{k=0,\dots,K-1} \sum_{i=1}^p \|\nabla_i f(X^k)\|_{(i)\star} \le \epsilon$, достаточно запустить алгоритм на

$$K = \left\lceil \frac{2\Delta^0 \sum_{i=1}^p L_i^0}{\epsilon^2} + \frac{2\Delta^0 L_{\max}^1}{\epsilon} \right\rceil; \tag{10}$$

2) Чтобы достичь точности

$$\min_{k=0,\dots,K-1} \sum_{i=1}^p \left[\frac{\frac{1}{L_i^1}}{\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}} \|\nabla_i f(X^k)\|_{(i)\star} \right] \le \varepsilon, \tag{11}$$

достаточно запустить алгоритм на

$$K = \left\lceil \frac{2\Delta^0 \left(\sum_{i=1}^p \frac{L_i^0}{(L_i^1)^2}\right)}{\varepsilon^2 \left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)^2} + \frac{2\Delta^0}{\varepsilon \left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)} \right\rceil, \tag{12}$$

где $\Delta^0 := f(X^0) - \inf_{X \in \delta} f(X)$ и $L_{\max}^1 := \max_{i=1,\dots,p} L_i^1$.

Отсюда следует несколько важных наблюдений.

Скорость сходимости. Сравним оценки (10) и (12). Из-за перевзвешивания норм компонент градиента в (11) скорости не вполне эквивалентны. Тем не менее обе используют веса, сумма которых равна p, что обеспечивает справедливое сравнение. Очевидно, $(1/p \sum_{j=1}^p 1/L_j^1)^{-1} \le L_{\max}^1$, так что второй член в (12) всегда не хуже своего аналога в (10). Сравнение же первых членов зависит от того, как соотносятся последовательности $\{L_i^0\}$ и $\{L_i^1\}$: если большим значениям L_i^0 обычно соответствуют меньшие значения L_i^1, то первый член в (10) лучше, чем в (12), а при положительной корреляции верно обратное. Действительно, в крайнем случае, когда $L_1^0 \ge \cdots \ge L_p^0$ и $L_1^1 \le \cdots \le L_p^1$ (или при обратном порядке), неравенство Чебышёва для сумм даёт

$$\frac{\sum_{i=1}^p \frac{L_i^0}{(L_i^1)^2}}{\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)^2} \ge \frac{\left(\frac{1}{p}\sum_{i=1}^p \frac{L_i^0}{L_i^1}\right)\left(\frac{1}{p}\sum_{i=1}^p \frac{1}{L_i^1}\right)}{\frac{1}{p}\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)^2} \ge \frac{\left(\frac{1}{p}\sum_{i=1}^p L_i^0\right)\left(\frac{1}{p}\sum_{i=1}^p \frac{1}{L_i^1}\right)}{\frac{1}{p}\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)} = \sum_{i=1}^p L_i^0.$$

Напротив, если обе последовательности $\{L_i^0\}$ и $\{L_i^1\}$ упорядочены одинаково (по возрастанию или убыванию), неравенство меняется на противоположное, и первый член (12) может быть точнее. При этом эмпирические данные, которые мы приводим в разделе 5, показывают, что на практике $L_i^0 \approx 0$ для всех слоёв, и тогда первые члены в (10) и (12) фактически обращаются в нуль. В этом случае (12) явно превосходит: худшая константа $L_{\max}^1$ заменяется средним гармоническим.

При $p = 1$ наши скорости совпадают с наилучшей известной сложностью поиска стационарной точки (L^0, L^1)-гладких функций, $\mathcal{O}(L^0\Delta^0/\epsilon^2 + L^1\Delta^0/\epsilon)$, установленной в [13] для градиентного метода. Хотя ни одна предшествующая работа не анализировала наш детерминированный подход при общей (L^0, L^1)-гладкости, существует анализ при классической L-гладкости, рассматривающий параметры как единый вектор. Анализ в [24] гарантирует сходимость за $K = \lceil 6L\Delta^0/\epsilon^2 \rceil$ итераций. Та же оценка появляется в [25] и [6] (при $\sigma^2 = 0$). Поскольку при $p = 1$ L-гладкость влечёт Предположение 1 с $L^1 = 0$ (Лемма 2), наши

скорости совпадают с этими прежними результатами с точностью до постоянного множителя. Таким образом, даже в гладком случае наши оценки столь же точны, как и выведенные специально для него.

Однако подлинная сила наших гарантий – в их более широкой применимости. Наш анализ гораздо более общий, чем предшествующие исследования, поскольку он выходит за рамки стандартной гладкости: допущение $L_i^1 > 0$ вводит дополнительные члены, обеспечивающие ускоренную сходимость, возможную при (L^0, L^1)-гладкости. Эта более богатая модель существенна для объяснения эмпирического ускорения методов вроде Muon и гораздо точнее отражает геометрию поверхностей потерь нейронных сетей. Действительно, как мы показываем в разделе 5, предположение, как правило, выполняется при $L_i^0 \approx 0$ и $L_i^1 > 0$.

Практические радиусы t_i^k. В отличие от прежних анализов [24-25, 6], требующих непрактично малых фиксированных радиусов, пропорциональных ϵ, наш подход обеспечивает динамическую адаптацию t_i^k к ландшафту потерь. Следовательно, t_i^k может принимать большие значения на ранних стадиях обучения, когда $\| \nabla_i f(X^k) \|_{(i)\star}$ велика, и естественно убывать по мере уменьшения нормы градиента. При эмпирически распространённом условии $L_i^0 \approx 0$ шаг упрощается до $t_i^k \approx 1/L_i^1$, давая радиус, значительно больший допускаемых прежними теориями. Важно, что, как мы показываем в разделе 5.1, эти теоретически выведенные шаги близко соответствуют оптимально настроенным скоростям обучения, обеспечивающим передовое качество в [6]. Такое сильное согласие подчёркивает, что принципиальный, математически обоснованный подход к выбору шага может значительно снизить зависимость от дорогостоящей ручной настройки.

Теперь установим скорости сходимости при послойном условии Поляка–Лоясевича (PŁ), вводимом в Предположении 2. Это свойство особенно актуально для сильно перепараметризованных нейронных сетей, так как, как было показано, оно отражает свойства их ландшафтов потерь [31].

Предположение 2 (Послойное условие Поляка–Лоясевича). Функция $f: \mathcal{S} \mapsto \mathbb{R}$ удовлетворяет послойному условию Поляка–Лоясевича (PŁ) с константой $\mu > 0$, т.е. для любого $X \in \mathcal{S}$

$$\sum_{i=1}^{p} \| \nabla_i f(X) \|_{(i)\star}^2 \geq 2\mu(f(X) - f^\star),$$

где $f^\star := \inf_{X \in \mathcal{S}} f(X) > -\infty$.

Предположение 2 сводится к стандартному условию PŁ [32] при векторизации параметров и использовании евклидовой нормы $\|\cdot\|_2$.

Теорема 2. Пусть выполнены Предположения 1 и 2 и зафиксировано $\varepsilon > 0$. Пусть $X^0, \dots, X^{K-1}$ – итерации детерминированного LMO-оптимизатора (Алгоритм 2), запущенного с $t_i^k = \frac{\|\nabla_i f(X^k)\|_{(i)\star}}{L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star}}$.

1) Если $L_i^1 \geq 0$, то для достижения точности $\min_{k=0,\dots,K-1} f(X^k) - f^\star \leq \epsilon$ достаточно запустить алгоритм на

$$K = \left\lceil \frac{\sum_{i=1}^p L_i^0 \Delta^0}{\mu\epsilon} + \frac{\sqrt{2} L_{\max}^1 \Delta^0}{\sqrt{\mu\epsilon}} \right\rceil;$$

2) Если $L_i^1 = 0$ для всех $i = 1, \dots, p$, то для достижения точности $f(X^K) - f^\star \leq \epsilon$ достаточно запустить алгоритм на

$$K = \left\lceil \frac{L_{\max}^0}{\mu} \log \frac{\Delta^0}{\epsilon} \right\rceil,$$

где $L_{\max}^0 := \max_{i=1,\dots,p} L_i^0$, $L_{\max}^1 := \max_{i=1,\dots,p} L_i^1$, $\Delta^0 := f(X^0) - f^\star$ и $f^\star := \inf_{X \in \mathcal{S}} f(X)$.

4.3. Стохастический случай

На практике вычисление полных градиентов часто неосуществимо из-за масштаба современных задач машинного обучения. Поэтому обратимся к практическому стохастическому варианту (Алгоритм 1), который оперирует зашумлёнными оценками градиента, доступными через стохастический градиентный оракул ∇f_ξ, $\xi \sim \mathcal{D}$.

Предположение 3 (ограниченная дисперсия). Стохастическая оценка градиента $\nabla f_\xi: \mathcal{S} \mapsto \mathcal{S}$ несмещённая и имеет ограниченную дисперсию. То есть $\mathbb{E}_{\xi\sim\mathcal{D}}[\nabla f_\xi(X)] = \nabla f(X)$ для всех $X \in \mathcal{S}$, и существует $\sigma \geq 0$ такое, что $\mathbb{E}_{\xi\sim\mathcal{D}}[\| \nabla_i f_\xi(X) - \nabla_i f(X) \|_2^2] \leq \sigma^2$ для всех $X \in \mathcal{S}$, $i = 1, \dots, p$.

Отметим, что выбор нормы в Предположении 3 не является ограничительным: в конечномерных пространствах все нормы эквивалентны, поэтому оценки дисперсии остаются справедливыми с точностью до постоянного множителя по сравнению с оценками, основанными на любой неевклидовой норме. Теперь установим основной результат о сходимости для стохастического случая. Здесь $\rho_i > 0$ для $i \in [p]$ обозначают константы эквивалентности норм, т.е. $\|X_i\|_{(i)\star} \leq \rho_i \|X_i\|_2$ для всех $X_i \in \mathcal{S}_i$.

Теорема 3. Пусть выполнены Предположения 1 и 3 и зафиксировано $\varepsilon > 0$. Пусть $X^0, \dots, X^{K-1}$ – итерации нашего стохастического подхода (Алгоритм 1), запущенного с $\beta^k = 1 - (k+1)^{-1/2}$, $t_i^k = t_i (k+1)^{-3/4}$ для некоторого $t_i > 0$, и $M_i^0 = \nabla_i f_{\xi^0}(X^0)$.

1) Если $L_i^1 = 0$, то

$$\min_{k=0,\dots,K-1} \sum_{i=1}^{p} t_i \, \mathbb{E}\left[\| \nabla_i f(X^k) \|_{(i)\star}\right] \leq \frac{\Delta^0}{K^{1/4}} + \frac{1}{K^{1/4}} \sum_{i=1}^{p} \left[\sigma \rho_i t_i \left(7 + 2\sqrt{2e^2}\log(K)\right) + L_i^0 t_i^2 \left(\frac{87}{2} + 14\log(K)\right)\right],$$

2) Если $L_i^1 \neq 0$, то для $t_i = \frac{1}{12 L_i^1}$ имеем

$$\min_{k=0,\dots,K-1} \sum_{i=1}^p \frac{1}{12L_i^1} \mathbb{E}\left[\| \nabla_i f(X^k) \|_{(i)\star}\right] \leq \frac{24^0}{K^{1/4}} + \frac{1}{K^{1/4}} \sum_{i=1}^p \left[\frac{\sigma\rho_i}{6L_i^1}\left(7 + 2\sqrt{2e^2}\log(K)\right) + \frac{L_i^0}{144 (L_i^1)^2}(87 + 28\log(K))\right] \quad (13)$$

где $\Delta^0 := f(X^0) - \inf_{X \in \mathcal{S}} f(X)$.

При $p = 1$ наша скорость в (13) восстанавливает сложность поиска стационарной точки (L^0, L^1)-гладких функций, установленную в [14] для нормализованного SGD с моментом. При $p \geq 1$, по сравнению с существующими гарантиями для этого класса методов, наша Теорема 3 действует при значительно более общем Предположении 1 и уникальным образом поддерживает обучение с большими, непостоянными шагами $t_i^k \propto k^{-3/4}$. Напротив, прежние анализы предписывают постоянные, исчезающе малые шаги $t_i^k \equiv t_i \propto K^{-3/4}$, привязанные к общему числу итераций K (см. табл. 1).

5. Эксперименты

Ниже мы приводим отдельные экспериментальные результаты для оптимизатора unScion – частного случая предложенного подхода (см. раздел 4.1). Дополнительные детали даны в Приложении B.

5.1. Обучение NanoGPT на FineWeb

В первой серии экспериментов мы стремимся проверить послойную (L^0, L^1)-гладкость (Предположение 1). С этой целью мы обучаем модель NanoGPT со 124M параметрами на

наборе данных FineWeb, используя два открытых GitHub-репозитория [27, 28]. Мы используем оптимизатор unScion, то есть наш подход с выбором норм как в (8). Мы заимствуем гиперпараметры из [6, табл. 7], отображая их значения $\gamma = 0{,}00036$, $\rho_2 = 50$ и $\rho_3 = 3000$ в наши обозначения следующим образом: $t_i^k \equiv \gamma\rho_2 = 0{,}018$ для $i = 1, \dots, p-1$ (что соответствует слоям трансформерных блоков) и $t_p^k \equiv \gamma\rho_3 = 1{,}08$ (эмбеддинги токенов и выходные проекции, в силу разделения весов). Мы устанавливаем число итераций «прогрева вниз» равным 0, чтобы скорости обучения оставались постоянными на протяжении всего обучения. Модель обучается в течение 5 000 итераций в соответствии с законами масштабирования Chinchilla для обеспечения вычислительно оптимального обучения. На рис. 2 мы строим оценку гладкости вдоль траектории как функцию индекса итерации k

$$\hat{L}_i[k] := \frac{\|\nabla_i f_{\xi^{k+1}}(X^{k+1}) - \nabla_i f_{\xi^k}(X^k)\|_{(i)\star}}{\|X_i^{k+1} - X_i^k\|_{(i)}} \tag{14}$$

для групп параметров из слоя эмбеддингов, а также 4-го и 8-го трансформерных блоков (схожие тенденции наблюдаются по всем блокам). Мы сравниваем это с аппроксимацией

$$\hat{L}_i^{\text{approx}}[k] := L_i^0 + L_i^1 \,\|\, \nabla_i f_{\xi^{k+1}}(X^{k+1}) \,\|_{(i)\star},$$

где $L_i^0, L_i^1 \geq 0$ подбираются для минимизации евклидовой ошибки между $\hat{L}_i[k]$ и $\hat{L}_i^{\text{approx}}[k]$ со штрафом hinge-типа за недооценку (см. Приложение B). Тесное согласие этих кривых означает, что Предположение 1 приблизительно выполняется вдоль траекторий обучения.

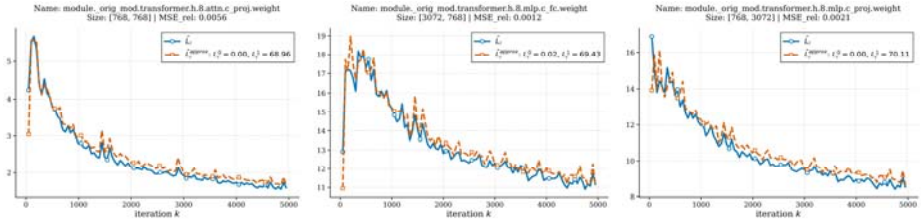

Рис. 2. Проверка Предположения 1 для 8-го трансформерного блока в NanoGPT-124M вдоль траекторий обучения.
Fig. 2. Verification of Assumption 1 for the 8th transformer block in NanoGPT-124M along the training trajectories

Далее мы оцениваем влияние масштабирующих множителей скорости обучения ρ_2 и ρ_3 на качество оптимизатора unScion. Для согласованности все прочие гиперпараметры фиксируются, как описано выше. В качестве базовой линии мы включаем результаты, полученные с оптимизатором AdamW при настройках гиперпараметров, указанных в [6, табл. 7]. Рис. 3 представляет (a) кривые валидации для обоих оптимизаторов при варьировании ρ_3 в unScion и (b) итоговую потерю на валидации для unScion при разных комбинациях ρ_2 и ρ_3. Наилучшее качество достигается при $\rho_2 = 50$ и $\rho_3 = 3000$, т.е. $t_i^k = 0{,}018$ для $i = 1, \dots, p-1$ и $t_p^k = 1{,}08$. Это подтверждает целесообразность неравномерного масштабирования по слоям с большими шагами для слоя эмбеддингов.

5.2. Обучение CNN на CIFAR-10

В этом эксперименте мы дополнительно проверяем послойную (L^0, L^1)-гладкость, обучая модель CNN на наборе данных CIFAR-10, следуя реализациям из двух открытых GitHub-репозиториев [33, 28]. Модель обучается с помощью оптимизатора unScion (9) с полнопакетными градиентами $\nabla_i f$, без момента и без затухания скорости обучения. Прочие гиперпараметры такие же, как в [6, табл. 10], за исключением того, что мы обучаем большее число эпох. Аналогично экспериментам с NanoGPT, обсуждавшимся в разделе 5.1, мы строим оценку (нестохастической) гладкости вдоль траектории $\hat{L}_i[k] := \|\nabla_i f(X^{k+1}) - \nabla_i f(X^k)\|_{(i)\star} /$

$\|X_i^{k+1} - X_i^k\|_{(i)}$ вместе с её аппроксимацией $\hat{L}_i^{\text{approx}}[k] := L_i^0 + L_i^1 \,\|\, \nabla_i f(X^{k+1}) \,\|_{(i)\star}$ для выбранных групп параметров. В этом эксперименте мы рассматриваем упрощённый вариант Предположения 1, полагая $L_i^0 = 0$, и оцениваем $L_i^1 \geq 0$ той же процедурой, что и в разделе 5.1. На рис. 4 представлены результаты, демонстрирующие, что Предположение 1 приблизительно выполняется вдоль траектории обучения. Когда это условие выполнено при $L_i^0 = 0$, Теорема 1 гарантирует сходимость при выборе шага $t_i^k \equiv t_i = 1/L_i^1$. В этом случае оценённые значения L_i^1 (показанные на рис. 4) равны $L_i^1 \approx 3$ для всех групп параметров, кроме весов классификационной головы X_p, где $L_p^1 \approx 0{,}03$. Эта разница примерно в два порядка величины обосновывает значительно больший радиус t_p^k, используемый для весов головы в настроенной конфигурации, приведённой в [6, табл. 10].

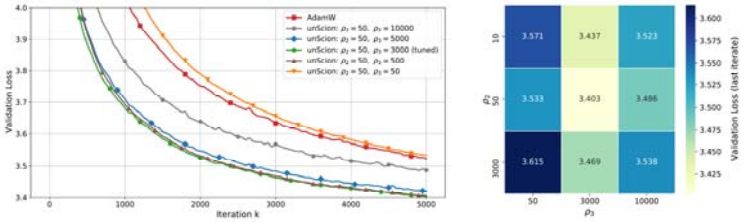

Рис. 3. (a) Кривые валидации для AdamW и unScion при разных значениях ρ_3; (b) тепловая карта потери на валидации на последней итерации unScion при разных комбинациях ρ_2 и ρ_3.
Fig. 3. (a) Validation curves for AdamW and unScion for different ρ_3; (b) heat-map of the final-iteration validation loss of unScion for different combinations of ρ_2 and ρ_3.

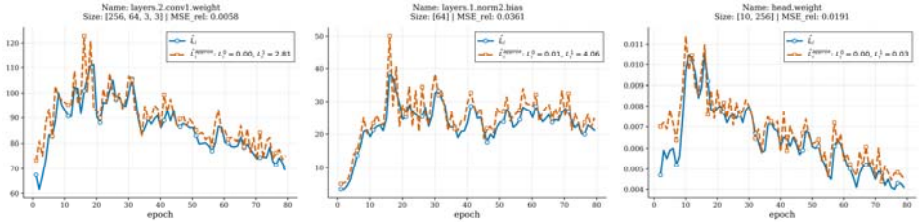

Рис. 4. Проверка Предположения 1 для разных групп параметров в CNN вдоль траекторий обучения unScion с полнопакетными градиентами.
Fig. 4. Verification of Assumption 1 for different parameter groups in a CNN along unScion training trajectories with full-batch gradients.

6. Заключение

В данной работе мы предложили обобщённый LMO-оптимизационный подход, охватывающий передовые оптимизаторы вроде Muon и Scion. Введя новое предположение о послойной (L^0, L^1)-гладкости, мы точно учли анизотропную геометрию современных глубоких сетей. Этот подход дал более точные и более общие гарантии сходимости, а также теоретические шаги, близко соответствующие эмпирически настроенным значениям. Экспериментальные результаты подтверждают, что наше предположение о гладкости выполняется на практике, успешно устраняя разрыв между теоретическим анализом и реальными реализациями оптимизаторов. В итоге наши результаты дают строгое, обладающее предсказательной силой основание для адаптивной послойной оптимизации в глубоком обучении.

Список литературы / References

[1]. Kingma D. P., Ba J. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015. DOI: 10.48550/arXiv.1412.6980.

[2]. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization. In International Conference on Learning Representations, 2019. DOI: 10.48550/arXiv.1711.05101.

[3]. Wilson A. C., Roelofs R., Stern M., Srebro N., Recht B. The marginal value of adaptive gradient methods in machine learning. Advances in Neural Information Processing Systems, 30, 2017. DOI: 10.48550/arXiv.1705.08292.

[4]. Zou D., Cao Y., Li Y., Gu Q. Understanding the generalization of Adam in learning neural networks with proper regularization. arXiv preprint, 2021. DOI: 10.48550/arXiv.2108.11371.

[5]. Jordan K., Jin Y., Boza V., You J., Cesista F., Newhouse L., Bernstein J. Muon: An optimizer for hidden layers in neural networks, 2024. Available at: https://kellerjordan.github.io/posts/muon/, accessed 13.07.2026.

[6]. Pethick T., Xie W., Antonakopoulos K., Zhu Z., Silveti-Falls A., Cevher V. Training Deep Learning Models with Norm-Constrained LMOs. arXiv preprint, 2025. DOI: 10.48550/arXiv.2502.07529.

[7]. Frank M., Wolfe P. An algorithm for quadratic programming. Naval Research Logistics Quarterly, 1956, vol. 3(1-2), pp. 95-110. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109, accessed 13.07.2026.

[8]. Pokutta S. The Frank-Wolfe algorithm: a short introduction. Jahresbericht der Deutschen Mathematiker-Vereinigung, vol. 126(1), pp. 3-35, 2024. DOI: 10.48550/arXiv.abs/2311.05313.

[9]. Liu J., Su J., Yao X., Jiang Z., Lai G., Du Y., Qin Y., Xu W., Lu E., Yan J. et al. Muon is scalable for LLM training. arXiv preprint, 2025. DOI: 10.48550/arXiv.2502.16982.

[10]. Crawshaw M., Liu M., Orabona F., Zhang W., Zhuang Z. Robustness to unbounded smoothness of generalized signSGD. Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 9955-9968. DOI: 10.48550/arXiv.2208.11195.

[11]. Zhang J., He T., Sra S., Jadbabaie A. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In International Conference on Learning Representations, 2020. DOI: 10.48550/arXiv.1905.11881.

[12]. Gorbunov E., Tupitsa N., Choudhury S., Aliev A., Richtárik P., Horváth S., Takáč M. Methods for convex (L0, L1)-smooth optimization: Clipping, acceleration, and adaptivity. In The Thirteenth International Conference on Learning Representations, 2025. DOI: 10.48550/arXiv.2409.14989.

[13]. Vankov D., Rodomanov A., Nedich A., Sankar L., Stich S. U. Optimizing (L0, L1)-Smooth Functions by Gradient Methods. In The Thirteenth International Conference on Learning Representations, 2025. DOI: 10.48550/arXiv.2410.10800.

[14]. Hübler F., Yang J., Li X., He N. Parameter-agnostic optimization under relaxed smoothness. In International Conference on Artificial Intelligence and Statistics. PMLR, 2024, pp. 4861-4869. DOI: 10.48550/arXiv.2311.03252.

[15]. Yu D., Jiang W., Wan Y., Zhang L. Mirror descent under generalized smoothness. arXiv preprint, 2025, DOI: 10.48550/arXiv.2502.00753.

[16]. Nesterov Y. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization, 2012, vol. 22(2), pp. 341-362. Available at: https://epubs.siam.org/doi/10.1137/100802001, accessed 13.07.2026.

[17]. Richtárik P., Takáč M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Mathematical Programming, 2014, vol. 144(1), pp. 1-38. DOI: 10.48550/arXiv.1107.2848.

[18]. Bernstein J., Wang Y.-X., Azizzadenesheli K., Anandkumar A. signSGD: Compressed optimisation for non-convex problems. In International Conference on Machine Learning. PMLR, 2018, pp. 560-569. DOI: 10.48550/arXiv.1802.04434.

[19]. Jiang R., Maladkar D., Mokhtari A. Convergence analysis of adaptive gradient methods under refined smoothness and noise assumptions. arXiv preprint, 2024. DOI: 10.48550/arXiv.2406.04592.

[20]. Liu Y., Pan R., Zhang T. AdaGrad under Anisotropic Smoothness, 2024. arXiv:2406.15244 [cs.LG]. DOI: 10.48550/arXiv.2406.15244.

[21]. Xie S., Mohamadi M. A., Li Z. Adam Exploits ℓ_∞-geometry of Loss Landscape via Coordinate-wise Adaptivity. arXiv preprint, 2024. DOI: 10.48550/arXiv.2410.08198.

[22]. Bernstein J., Newhouse L. Old Optimizer, New Norm: An Anthology. In OPT 2024: Optimization for Machine Learning, 2024. DOI: 10.48550/arXiv.2409.20325.

[23]. Jaggi M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In International Conference on Machine Learning. PMLR, 2013, pp. 427-435.

[24]. Kovalev D. Understanding Gradient Orthogonalization for Deep Learning via Non-Euclidean Trust-Region Optimization, 2025. arXiv:2503.12645 [cs.LG]. DOI: 10.48550/arXiv.2503.12645.

[25]. Li J., Hong M. A Note on the Convergence of Muon and Further, 2025. arXiv:2502.02900 [math.OC]. DOI: 10.48550/arXiv.2502.02900.

[26]. Shah I., Polloreno A. M., Stratos K., Monk P., Chaluvaraju A., Hojel A., Ma A., Thomas A., Tanwer A., Shah D. J. et al. Practical Efficiency of Muon for Pretraining. arXiv preprint, 2025. DOI: 10.48550/arXiv.

[27]. Jordan K., Bernstein J., Rappazzo B., Vlado B., Jiacheng Y., Cesista F., Koszarsky B. Modded-nanoGPT: Speedrunning the nanoGPT Baseline. GitHub repository, 2024. Available at: https://github.com/KellerJordan/modded-nanogpt, accessed 13.07.2026.

[28]. Pethick T., Xie W., Antonakopoulos K., Zhu Z., Silveti-Falls A., Cevher V. Scion. GitHub repository, 2025. Available at: https://github.com/LIONS-EPFL/scion.git, accessed 13.07.2026.

[29]. Yu A. W., Huang L., Lin Q., Salakhutdinov R., Carbonell J. Block-Normalized Gradient Method: An Empirical Study for Training Deep Neural Network, 2018. Available at: https://openreview.net/forum?id=ry831QWAb, accessed 13.07.2026.

[30]. Balles L., Pedregosa F., Roux N. L. The Geometry of Sign Gradient Descent, 2020. arXiv:2002.08056 [cs.LG]. DOI: 10.48550/arXiv.2002.08056.

[31]. Liu C., Zhu L., Belkin M. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. Applied and Computational Harmonic Analysis, 59, 2022. DOI: 10.1016/j.acha.2021.12.009. Available at: https://arxiv.org/abs/2003.00307, accessed 13.07.2026.

[32]. Karimi H., Nutini J., Schmidt M. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Łojasiewicz Condition, 2020. arXiv:1608.04636 [cs.LG]. DOI: 10.48550/arXiv.1608.04636.

[33]. Jordan K. Cifar-10 Airbench. GitHub repository, 2024. Available at: https://github.com/KellerJordan/cifar10-airbench, accessed 13.07.2026.

Информация об авторах / Information about authors

Артем Александрович РЯБИНИН – специалист лаборатории фундаментальных исследований искусственного интеллекта Московского независимого исследовательского института искусственного интеллекта. Научные интересы включают в себя теоретическую оптимизацию, компьютерное зрение и диффузионные модели.

Artem Aleksandrovich RIABININ – specialist at the Basic Research of Artificial Intelligence Laboratory (BRAIn Lab) of the Moscow Independent Research Institute of Artificial Intelligence. Research interests include theoretical optimization, computer vision, and diffusion models.

Приложение

A. Доказательства

A.1. Вспомогательные леммы

Лемма 1. Пусть $f:\mathcal{S}\mapsto\mathbb{R}$ удовлетворяет Предположению 1. Тогда для любых $X,Y\in\mathcal{S}$ имеем

$$|f(Y)-f(X)-\langle\nabla f(X),Y-X\rangle|\le\sum_{i=1}^{p}\frac{L_i^0+L_i^1\,\|\,\nabla_i f(X)\,\|_{(i)\star}}{2}\,\|\,Y_i-X_i\,\|_{(i)}^2.$$

Доказательство. Для всех $X,Y\in\mathcal{S}$ имеем

$$f(Y)=f(X)+\int_0^1\langle\nabla f(X+\tau(Y-X)),Y-X\rangle\,d\tau$$

$$=f(X)+\langle\nabla f(X),Y-X\rangle+\int_0^1\langle\nabla f(X+\tau(Y-X))-\nabla f(X),Y-X\rangle\,d\tau.$$

Следовательно, используя неравенство Коши–Буняковского и Предположение 1, получаем

$$\begin{aligned}&|f(Y)-f(X)-\langle\nabla f(X),Y-X\rangle|\\&\le\left|\int_0^1\sum_{i=1}^{p}\langle\nabla_i f(X+\tau(Y-X))-\nabla_i f(X),Y_i-X_i\rangle_{(i)}\,d\tau\right|\\&\le\int_0^1\sum_{i=1}^{p}\left|\langle\nabla_i f(X+\tau(Y-X))-\nabla_i f(X),Y_i-X_i\rangle_{(i)}\right|d\tau\\&\le\int_0^1\sum_{i=1}^{p}\|\nabla_i f(X+\tau(Y-X))-\nabla_i f(X)\|_{(i)\star}\,\|\,Y_i-X_i\,\|_{(i)}\,d\tau\\&\le\int_0^1\sum_{i=1}^{p}\tau\left(L_i^0+L_i^1\,\|\,\nabla_i f(X)\,\|_{(i)\star}\right)\|\,Y_i-X_i\,\|_{(i)}^2\,d\tau\\&=\sum_{i=1}^{p}\frac{L_i^0+L_i^1\,\|\,\nabla_i f(X)\,\|_{(i)\star}}{2}\,\|\,Y_i-X_i\,\|_{(i)}^2.\end{aligned}$$

■

Лемма 2. Предположим, что f является L-гладкой относительно нормы, определённой для $X=[X_1,\dots,X_p]\in\mathcal{S}$ как

$$\|X\|_{\max}:=\max\{\|X_1\|_{(1)},\dots,\|X_p\|_{(p)}\},$$

то есть для всех $X=[X_1,\dots,X_p]$ и $Y=[Y_1,\dots,Y_p]$ с $X_i,Y_i\in\mathcal{S}_i$,

$$\|\nabla f(X)-\nabla f(Y)\|_{\max\star}\le L\|X-Y\|_{\max}.$$

Тогда Предположение 1 выполняется с $L_i^0\le L$ и $L_i^1=0$ для всех $i=1,\dots,p$.

Доказательство. L-гладкость и определение нормы дают

$$\sum_{i=1}^{p}\|\,\nabla_i f(X)-\nabla_i f(Y)\,\|_{(i)\star}\le L\max\left\{\|X_1-Y_1\|_{(1)},\dots,\|X_p-Y_p\|_{(p)}\right\}$$

для всех $X,Y\in\mathcal{S}$. В частности, выбирая $X=[X_1,\dots,X_p]$ и $Y=[X_1,\dots,X_{j-1},Y_j,X_{j+1},\dots X_p]$, имеем

$$\|\,\nabla_j f(X)-\nabla_j f(Y)\,\|_{(j)\star}\le\sum_{i=1}^{p}\|\,\nabla_i f(X)-\nabla_i f(Y)\,\|_{(i)\star}\le L\|X_j-Y_j\|_{(j)}$$

для любого $j\in\{1,\dots,p\}$, что доказывает утверждение.

■

Лемма 3. Предположим, что $x_1,\dots,x_p,y_1,\dots,y_p\in\mathbb{R}$, $\max_{i\in[p]}|x_i|>0$ и $z_1,\dots,z_p>0$. Тогда

$$\sum_{i=1}^{p}\frac{y_i^2}{z_i}\ge\frac{\left(\sum_{i=1}^{p}x_i\,y_i\right)^2}{\sum_{i=1}^{p}z_i\,x_i^2}.$$

Доказательство. Неравенство Коши–Буняковского даёт

$$\left(\sum_{i=1}^{p}x_i\,y_i\right)^2=\left(\sum_{i=1}^{p}\frac{y_i}{\sqrt{z_i}}\sqrt{z_i}x_i\right)^2\le\left(\sum_{i=1}^{p}\frac{y_i^2}{z_i}\right)\left(\sum_{i=1}^{p}z_i\,x_i^2\right).$$

Переставляя члены, получаем результат.

■

Лемма 4 (Техническая лемма 10 из [14]). Пусть $q\in(0,1)$, $p\ge0$ и $p\ge q$. Далее, пусть $a,b\in\mathbb{N}_{\ge2}$ с $a\le b$. Тогда

$$\sum_{k=a-1}^{b-1}(1+k)^{-p}\prod_{\tau=a-1}^{k}(1-(\tau+1)^{-q})\le(a-1)^{q-p}\exp\left(\frac{a^{1-q}-(a-1)^{1-q}}{1-q}\right).$$

Лемма 5 (Техническая лемма 11 из [14]). Пусть $t>0$ и для $k\in\mathbb{N}_{\ge0}$ положим $\beta^k=1-(k+1)^{-1/2}$, $t^k=t(k+1)^{-3/4}$, $t>0$. Тогда для всех $K\in\mathbb{N}_{\ge1}$ выполняются следующие неравенства:

(i) $$\sum_{k=0}^{K-1}t^k\sqrt{\sum_{\tau=0}^{k}(1-\beta^\tau)^2\prod_{\kappa=\tau+1}^{k}(\beta^\kappa)^2}\le t\left(7/2+\sqrt{2e^2}\log(K)\right);$$

(ii) $$\sum_{k=0}^{K-1}t^k\sum_{\tau=1}^{k}t^\tau\prod_{\kappa=\tau}^{k}\beta^\kappa\le7t^2(3+\log(K)).$$

Доказательство. Это прямое следствие Леммы 11 из [14]. Чтобы получить (ii), достаточно перейти к пределу при $L^1\to0$ в утверждении (ii) части (b).

■

A.2. Доказательство Теоремы 1

Доказательство. Начнём с результата, полученного в Лемме 1 при $X=X^k$ и $Y=X^{k+1}$:

$$\begin{aligned}f(X^{k+1})&\le f(X^k)+\langle\nabla f(X^k),X^{k+1}-X^k\rangle+\sum_{i=1}^{p}\frac{L_i^0+L_i^1\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}}{2}\,\|\,X_i^k-X_i^{k+1}\,\|_{(i)}^2\\&=f(X^k)+\sum_{i=1}^{p}\left[\left\langle\nabla_i f(X^k),X_i^{k+1}-X_i^k\right\rangle_{(i)}+\frac{L_i^0+L_i^1\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}}{2}\,\|\,X_i^k-X_i^{k+1}\,\|_{(i)}^2\right].\end{aligned}$$

Правило обновления (5) и определение двойственной нормы $\|\cdot\|_{(i)\star}$ дают $\|\,X_i^k-X_i^{k+1}\,\|_{(i)}^2\le\left(t_i^k\right)^2$ и

$$\begin{aligned}\left\langle\nabla_i f(X^k),X_i^{k+1}-X_i^k\right\rangle_{(i)}&=\left\langle\nabla_i f(X^k),\mathrm{LMO}_{\mathcal{B}_i^k}(\nabla_i f(X^k))-X_i^k\right\rangle_{(i)}\\&=-t_i^k\max_{\|X_i\|_{(i)}\le1}\langle\nabla_i f(X^k),X_i\rangle_{(i)}=-t_i^k\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}.\end{aligned}$$

Следовательно,

$$f(X^{k+1})\le f(X^k)+\sum_{i=1}^{p}\left[-t_i^k\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}+\frac{L_i^0+L_i^1\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}}{2}\left(t_i^k\right)^2\right].$$

Теперь, выбирая $t_i^k = \frac{\|\nabla_i f(X^k)\|_{(i)\star}}{L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star}}$, что минимизирует правую часть последнего неравенства, получаем неравенство убывания

$$f(X^{k+1}) \le f(X^k) - \sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|^2_{(i)\star}}{2\left(L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star}\right)}. \tag{15}$$

Суммируя члены, получаем

$$\begin{aligned} \sum_{k=0}^{K-1} \sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|^2_{(i)\star}}{2\left(L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star}\right)} &\le \sum_{k=0}^{K-1} \left(f(X^k) - f(X^{k+1})\right) \\ &= f(X^0) - f(X^K) \le f(X^0) - \inf_{X \in \mathcal{S}} f(X) =: \Delta^0. \end{aligned} \tag{16}$$

Теперь анализ может пойти двумя путями.

1) Оценивая L_i^1 сверху через $L_{\max}^1 := \max\limits_{i=1,\dots,p} L_i^1$ в (16), получаем

$$\sum_{k=0}^{K-1} \sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|^2_{(i)\star}}{2\left(L_i^0 + L_{\max}^1 \|\nabla_i f(X^k)\|_{(i)\star}\right)} \le \Delta^0. \tag{17}$$

Теперь, применяя Лемму 3 с $x_i = 1$, $y_i = \| \nabla_i f(X^k) \|_{(i)\star}$ и $z_i = 2\left(L_i^0 + L_{\max}^1 \|\nabla_i f(X^k)\|_{(i)\star}\right)$, получаем

$$\begin{aligned} \phi\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right) &= \frac{\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right)^2}{2\left(\sum_{i=1}^p L_i^0 + L_{\max}^1 \sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right)} \\ &\le \sum_{i=1}^p \frac{\| \nabla_i f(X^k) \|^2_{(i)\star}}{2\left(L_i^0 + L_{\max}^1 \| \nabla_i f(X^k) \|_{(i)\star}\right)}, \end{aligned}$$

где $\phi(t) := \frac{t^2}{2\left(\sum_{i=1}^p L_i^0 + L_{\max}^1 t\right)}$. Объединяя последнее неравенство с (17) и используя то, что ϕ возрастает, получаем

$$K\phi\left(\min_{k=0,\dots,K-1} \sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right) \le \sum_{k=0}^{K-1} \phi\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right) \le \Delta^0, \tag{18}$$

и, следовательно, $\min\limits_{k=0,\dots,K-1} \sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star} \le \phi^{-1}\left(\frac{\Delta^0}{K}\right)$, где ϕ^{-1} – обратная функция (существующая, так как ϕ возрастает). Поэтому для достижения точности $\min\limits_k \sum_{i=1}^p \|\nabla_i f(X^k)\|_{(i)\star} \le \epsilon$ достаточно выбрать число итераций равным

$$K = \left\lceil \frac{\Delta^0}{\phi(\epsilon)} \right\rceil = \left\lceil \frac{2\sum_{i=1}^p L_i^0 \Delta^0}{\epsilon^2} + \frac{2L_{\max}^1 \Delta^0}{\epsilon} \right\rceil.$$

2) В качестве альтернативы можно начать с неравенства (16) и применить Лемму 3 с $x_i = 1/L_i^1$, $y_i = \|\nabla_i f(X^k)\|_{(i)\star}$ и $z_i = 2(L_i^0 + L_i^1 \|\nabla_i f(X^k)\|_{(i)\star})$, чтобы получить

$$\Delta^0 \ge \sum_{k=0}^{K-1} \sum_{i=1}^p \frac{\| \nabla_i f(X^k) \|^2_{(i)\star}}{2\left(L_i^0 + L_i^1 \| \nabla_i f(X^k) \|_{(i)\star}\right)}$$

$$\ge \sum_{k=0}^{K-1} \frac{\left(\sum_{i=1}^p \frac{1}{L_i^1} \|\nabla_i f(X^k)\|_{(i)\star}\right)^2}{2\left(\sum_{i=1}^p \frac{L_i^0}{(L_i^1)^2} + \sum_{i=1}^p \frac{1}{L_i^1} \|\nabla_i f(X^k)\|_{(i)\star}\right)} = \sum_{k=0}^{K-1} \psi\left(\sum_{i=1}^p \frac{1}{L_i^1} \|\nabla_i f(X^k)\|_{(i)\star}\right),$$

где $\psi(t) := \frac{t^2}{2\left(\sum_{i=1}^p \frac{L_i^0}{(L_i^1)^2} + t\right)}$.

Поскольку функция ψ возрастает при $t > 0$, ψ^{-1} существует. Отсюда $\Delta^0 \ge K\psi\left(\min\limits_k \sum_{i=1}^p \frac{1}{L_i^1} \|\nabla_i f(X^k)\|_{(i)\star}\right)$, и, следовательно,

$$\min_{k=0,\dots,K-1} \sum_{i=1}^p \frac{1}{L_i^1} \|\nabla_i f(X^k)\|_{(i)\star} \le \psi^{-1}\left(\frac{\Delta^0}{K}\right).$$

Это, в свою очередь, означает, что для достижения точности $\min\limits_k \sum_{i=1}^p \left[\frac{\frac{1}{L_i^1}}{\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}} \|\nabla_i f(X^k)\|_{(i)\star}\right] \le \varepsilon$ достаточно запустить алгоритм на

$$K = \left\lceil \frac{\Delta^0}{\psi\left(\varepsilon\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)\right)} \right\rceil = \left\lceil \frac{2\Delta^0\left(\sum_{i=1}^p \frac{L_i^0}{(L_i^1)^2}\right)}{\varepsilon^2\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)^2} + \frac{2\Delta^0}{\varepsilon\left(\frac{1}{p}\sum_{j=1}^p \frac{1}{L_j^1}\right)} \right\rceil$$ итераций.

■

A.3. Доказательство Теоремы 2

Доказательство. Рассмотрим два сценария: (1) общий случай с произвольным $L_i^1 \ge 0$ и (2) $L_i^1 = 0$ для всех $i = 1, \dots, p$.

Случай 1: $L_i^1 \ge 0$**.** Начнём с тех же шагов, что и в доказательстве Теоремы 1. Из (18) имеем $\sum_{k=0}^{K-1} \phi\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right) \le \Delta^0$, где $\phi(t) := \frac{t^2}{2\left(\sum_{i=1}^p L_i^0 + L_{\max}^1 t\right)}$. Теперь, используя Предположение 2, получаем

$$\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right)^2 \ge \sum_{i=1}^p \| \nabla_i f(X^k) \|^2_{(i)\star} \ge 2\mu(f(X^k) - f^\star).$$

Следовательно, поскольку ϕ – возрастающая функция,

$$\begin{aligned} K\phi\left(\sqrt{2\mu}\sqrt{f(X^{k^\star}) - f^\star}\right) &\le \sum_{k=0}^{K-1} \phi\left(\sqrt{2\mu}\sqrt{f(X^k) - f^\star}\right) \\ &\le \sum_{k=0}^{K-1} \phi\left(\sum_{i=1}^p \| \nabla_i f(X^k) \|_{(i)\star}\right) \le \Delta^0, \end{aligned}$$

где $k^\star := \operatorname{argmin}_{k=0,\dots,K-1} f(X^k) - f^\star$. Обозначая обратную функцию через ϕ^{-1}, получаем $\sqrt{2\mu}\sqrt{f(X^{k^\star}) - f^\star} \le \phi^{-1}(\Delta^0/K) \le \sqrt{2\mu\epsilon}$. Поэтому для достижения точности $f(X^{k^\star}) - f^\star \le \epsilon$ достаточно выбрать число итераций

$$K = \left\lceil \frac{\Delta^0}{\phi\left(\sqrt{2\mu\epsilon}\right)} \right\rceil = \left\lceil \frac{\sum_{i=1}^p L_i^0 \Delta^0}{\mu\epsilon} + \frac{\sqrt{2}L_{\max}^1 \Delta^0}{\sqrt{\mu\epsilon}} \right\rceil.$$

Случай 2: $L_i^1 = 0$**.** Неравенство (15) из доказательства Теоремы 1 при $L_i^1 = 0$ даёт $f(X^{k+1}) \le f(X^k) - \sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|^2_{(i)\star}}{2L_i^0}$. Используя то, что

$$\sum_{i=1}^p \frac{\| \nabla_i f(X^k) \|^2_{(i)\star}}{2L_i^0} \ge \frac{1}{2\max\limits_{j=1,\dots,p} L_j^0} \sum_{i=1}^p \| \nabla_i f(X^k) \|^2_{(i)\star}$$

вместе с Предположением 2, получаем $f(X^{k+1}) \le f(X^k) - \frac{\mu}{L_{\max}^0}(f(X^k) - f^\star)$.

Оставшаяся часть доказательства следует из простого наблюдения

$$\log\left(\frac{\Delta_0}{\epsilon}\right) \le k\frac{\mu}{L^0_{\max}} \le k\log\left(\frac{1}{1-\frac{\mu}{L^0_{\max}}}\right).$$

■

A.4. Доказательство Теоремы 3

Доказательство. Снова начнём с результата Леммы 1 при $X = X^k$ и $Y = X^{k+1}$, получая

$$\begin{aligned} f(X^{k+1}) &\le f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k\rangle + \sum_{i=1}^{p} \frac{L_i^0 + L_i^1 \,\|\, \nabla_i f(X^k)\,\|_{(i)\star}}{2} \|\, X_i^k - X_i^{k+1}\,\|^2_{(i)} \\ &= f(X^k) + \sum_{i=1}^{p} [\,\langle M_i^k, X_i^{k+1} - X_i^k\rangle_{(i)} + \langle \nabla_i f(X^k) - M_i^k, X_i^{k+1} - X_i^k\rangle_{(i)}\,] \\ &\quad + \sum_{i=1}^{p} \frac{L_i^0 + L_i^1 \,\|\, \nabla_i f(X^k)\,\|_{(i)\star}}{2} \|\, X_i^k - X_i^{k+1}\,\|^2_{(i)}. \end{aligned}$$

Применяя неравенство Коши–Буняковского и используя правило обновления (2) и определение двойственной нормы (которые дают $\|\, X_i^k - X_i^{k+1}\,\|^2_{(i)} \le (t_i^k)^2$ и $\langle M_i^k, X_i^{k+1} - X_i^k\rangle = -t_i^k \,\|\, M_i^k\,\|_{(i)\star}$), получаем

$$\begin{aligned} f(X^{k+1}) &\le f(X^k) + \sum_{i=1}^{p} \left[-t_i^k \,\|\, \nabla_i f(X^k)\,\|_{(i)\star} + 2t_i^k \,\|\, M_i^k - \nabla_i f(X^k)\,\|_{(i)\star}\right] \\ &\quad + \sum_{i=1}^{p} \frac{L_i^0 + L_i^1 \,\|\, \nabla_i f(X^k)\,\|_{(i)\star}}{2} \left(t_i^k\right)^2. \end{aligned}$$

Беря математическое ожидание и телескопируя, получаем

$$\begin{aligned} \textstyle\sum_{i=1}^{p}\sum_{k=0}^{K-1} t_i^k\, \mathbb{E}[\,\|\, \nabla_i f(X^k)\,\|_{(i)\star}] &\le \Delta^0 + \textstyle\sum_{i=1}^{p}[\,2\sum_{k=0}^{K-1} t_i^k\, \mathbb{E}\left[\|\nabla_i M_i^k - \nabla_i f(X^k)\|_{(i)\star}\right] \\ &\quad + \textstyle\sum_{k=0}^{K-1} \frac{L_i^0}{2}\left(t_i^k\right)^2 + \sum_{k=0}^{K-1} \frac{L_i^1}{2}\mathbb{E}[\,\|\, \nabla_i f(X^k)\,\|_{(i)\star}]\left(t_i^k\right)^2], \end{aligned} \tag{19}$$

где $\Delta^0 := f(X^0) - \inf\limits_{X\in\mathcal{S}} f(X)$. Теперь, вдохновляясь анализом из [14], введём обозначения:

$$\mu_i^k := M_i^k - \nabla_i f(X^k),\ \gamma_i^k := \nabla_i f_{\xi^k}(X^k) - \nabla_i f(X^k),\ \alpha^k = 1-\beta^k,\ \beta^{a:b} := \textstyle\prod_{k=a}^{b}\beta^k \text{ и}$$
$$S_i^k := \nabla_i f(X^{k-1}) - \nabla_i f(X^k).$$

Тогда правило обновления момента можно переписать как $M_i^k = \nabla_i f(X^k) + \alpha^k\gamma_i^k + \beta^k S_i^k + \beta^k\mu_i^{k-1}$, что даёт $\mu_i^k = \sum_{\tau=0}^{k}\beta^{(\tau+1):k}\,\alpha^\tau\gamma_i^\tau + \sum_{\tau=1}^{k}\beta^{\tau:k}\,S_i^\tau$ (используя $M_i^0 = \nabla_i f_{\xi^0}(X^0)$ и $\beta^0 = 0$). Таким образом,

$$\mathbb{E}\left[\|\nabla_i M_i^k - \nabla_i f(X^k)\|_{(i)\star}\right] \le \rho_i \sqrt{\sum_{\tau=0}^{k}(\beta^{(\tau+1):k}\alpha^\tau)^2\,\mathbb{E}\left[\|\gamma_i^\tau\|_2^2\right]} + \sum_{\tau=1}^{k}\beta^{\tau:k}\,\mathbb{E}\left[\|S_i^\tau\|_{(i)\star}\right],$$

где мы использовали неравенство Йенсена и тот факт, что для всех $q < l$ выполняется $\mathbb{E}[(\gamma_i^l)^\top\gamma_i^q] = 0$.

Используя Предположения 1 и 3 ($\mathbb{E}[\|\,\gamma_i^\tau\,\|_2^2] \le \sigma^2$ и $\|\,S_i^\tau\,\|_{(i)\star} \le (L_i^0 + L_i^1\,\|\,\nabla_i f(X^\tau)\,\|_{(i)\star})t_i^\tau$), получаем

$$\begin{aligned} \mathbb{E}\left[\|\nabla_i M_i^k - \nabla_i f(X^k)\|_{(i)\star}\right] &\le \sigma\rho_i \sqrt{\sum_{\tau=0}^{k}(\beta^{(\tau+1):k}\alpha^\tau)^2} + L_i^0\sum_{\tau=1}^{k}\beta^{\tau:k}\,t_i^\tau \\ &\quad + L_i^1\sum_{\tau=1}^{k}\beta^{\tau:k}\,t_i^\tau\,\mathbb{E}\left[\|\,\nabla_i f(X^\tau)\,\|_{(i)\star}\right]. \end{aligned}$$

Объединяя последнее неравенство с (19), получаем

$$\begin{aligned} \textstyle\sum_{i=1}^{p}\sum_{k=0}^{K-1} t_i^k\,\mathbb{E}[\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}] \le \Delta^0 + \sum_{i=1}^{p}[\,&(2\sigma\rho_i\sum_{k=0}^{K-1} t_i^k\sqrt{\textstyle\sum_{\tau=0}^{k}(\beta^{(\tau+1):k}\alpha^\tau)^2}) + (2L_i^0\sum_{k=0}^{K-1} t_i^k\sum_{\tau=1}^{k}\beta^{\tau:k}\,t_i^\tau) \\ &+(2L_i^1\sum_{k=0}^{K-1} t_i^k\sum_{\tau=1}^{k}\beta^{\tau:k}\,t_i^\tau\mathbb{E}[\,\|\,\nabla_i f(X^\tau)\,\|_{(i)\star}]) + (\tfrac{L_i^0}{2}\textstyle\sum_{k=0}^{K-1}(t_i^k)^2) \\ &+\tfrac{L_i^1}{2}\textstyle\sum_{k=0}^{K-1}(t_i^k)^2\mathbb{E}[\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}]], \end{aligned} \tag{20}$$

где четыре отмеченных члена в порядке их появления обозначены I_1, I_2, I_3, I_4. Оценим теперь сверху каждый из них. Используя Лемму 5, получаем $I_1 \le \sigma\rho_i t_i(7 + 2\sqrt{2e^2}\log(K))$ и $I_2 \le 14L_i^0 t_i^2(3 + \log(K))$. Переставляя суммы и используя Лемму 4 с $a = \tau + 1$, $b = K$, $p = 3/4$ и $q = 1/2$, имеем $I_3 \le 2e^{2(\sqrt{2}-1)}L_i^1\sum_{k=0}^{K-1} t_i^k\, t_i\mathbb{E}[\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}]$. Наконец,

$$I_4 = \frac{L_i^0}{2}\sum_{k=0}^{K-1}\left(t_i^k\right)^2 \le \frac{L_i^0}{2}t_i^2\left(1 + \int_1^\infty \frac{1}{z^{3/2}}\,dz\right) = \frac{3L_i^0}{2}t_i^2.$$

Объединяя верхние оценки для I_i с (20), используя $t_i^k = t_i(1+k)^{-3/4} \le t_i$ и обозначая $C := 2e^{2(\sqrt{2}-1)} + \frac{1}{2} \le 5{,}1$, и рассматривая два случая, приходим к утверждению теоремы.

Случай 1: $L_i^1 = 0$**.** В этом случае

$$\begin{aligned} \min_{k=0,\dots,K-1}\sum_{i=1}^{p} t_i\,\mathbb{E}[\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}] &\le \frac{\Delta^0}{K^{1/4}} + \frac{1}{K^{1/4}}\sum_{i=1}^{p}[\,\sigma\rho_i t_i\left(7 + 2\sqrt{2e^2}\log(K)\right) + L_i^0 t_i^2(87/2 + 14\log(K))]. \end{aligned}$$

Случай 2: $L_i^1 \ne 0$**.** Выберем $t_i = \frac{1}{12L_i^1}$. Тогда

$$\begin{aligned} \min_{k=0,\dots,K-1}\sum_{i=1}^{p}\frac{1}{12L_i^1}\,\mathbb{E}[\,\|\,\nabla_i f(X^k)\,\|_{(i)\star}] &\le \frac{2\Delta^0}{K^{1/4}} + \frac{1}{K^{1/4}}\sum_{i=1}^{p}[\,\frac{\sigma\rho_i}{6L_i^1}\left(7 + 2\sqrt{2e^2}\log(K)\right) + \frac{L_i^0}{144(L_i^1)^2}(87 + 28\log(K))]. \end{aligned}$$

■

B. Детали экспериментов

Для LMO-методов мы вычисляем приближённые решения задачи LMO (неточный оракул) с помощью итерации Ньютона–Шульца, когда аналитическое решение недоступно (например, для эффективных обновлений сингулярного типа – Singular Value Decomposition, SVD), следуя подходу, предложенному в работе [5]. Этот метод даёт вычислительно эффективную аппроксимацию необходимой ортогонализации, сохраняя при этом поведение сходимости всего алгоритма.

Чтобы минимизировать евклидову ошибку между истинным значением $\hat{L}_i[k]$ и его аппроксимацией $\hat{L}_i^{\text{approx}}[k]$, штрафуя при этом недооценку, мы вводим штрафной член hinge-типа. В частности, мы подбираем L_i^0 и L_i^1, минимизируя функцию потерь

$$\mathcal{L}_i(L_i^0, L_i^1) := \sum_{k=0}^{K-1}\left(\hat{L}_i[k] - \hat{L}_i^{\text{approx}}[k]\right)^2 + \lambda \sum_{k=0}^{K-1} \max\left(0, \hat{L}_i[k] - \hat{L}_i^{\text{approx}}[k]\right)^2. \qquad (21)$$

Первый член $\mathcal{L}_i$ отражает стандартную евклидову (квадратичную) ошибку, тогда как второй член вводит дополнительный штраф, пропорциональный величине недооценки (когда $\hat{L}_i[k] > \hat{L}_i^{\text{approx}}[k]$). Гиперпараметр $\lambda \geq 0$ управляет силой этого штрафа.

Все эксперименты для модели NanoGPT проводятся с использованием платформы PyTorch с модулем Distributed Data Parallel (DDP) на 4-х графических процессорах GPU NVIDIA A100 (по 40 ГБ). Для экспериментов со сверточными нейронными сетями (CNN) обучение выполняется на одном процессоре GPU NVIDIA A100 (40 ГБ). Конвейеры обучения и оценки реализованы с использованием открытых кодовых баз [33, 27, 28], при этом все модификации чётко задокументированы и должным образом помечены, где это применимо.