

Аннотация

В современном мире машинное обучение невозможно представить без использования больших обучающих выборок и моделей. Это обусловило широкое применение стохастических методов обучения, таких как SGD. Несмотря на простоту, SGD обладает слабыми теоретическими гарантиями сходимости, связанными с неубывающей дисперсией. Данная проблема может быть частично устранена с помощью модификаций, таких как SARAH. Однако эти методы требуют периодического вычисления полного градиента, что может быть затратным по времени. В данной работе были рассмотрены варианты алгоритмов с уменьшением дисперсии, не предполагающие необходимость вычисления полного градиента. Для повышения эффективности по памяти и исключения этих вычислений были использованы два ключевых подхода: эвристика перемешивания и идея, лежащая в основе методов SAG/SAGA. В результате были улучшены существующие оценки для алгоритмов с уменьшением дисперсии без вычисления полного градиента. В случае невыпуклой целевой функции полученная оценка совпадает с классическими методами на основе перемешивания, а для сильно выпуклой задачи достигается улучшение. Проведён всесторонний теоретический анализ, а также представлены масштабные экспериментальные результаты, подтверждающие эффективность и практическую применимость предложенных методов в задачах обучения на больших данных.

Содержание

1 Введение	4
2 Обзор литературы	7
3 Постановка задачи	10
4 Основные результаты	11
4.1 Аппроксимация полного градиента	11
4.2 SARAH без полного градиента	12
4.2.1 Невыпуклая постановка	13
4.2.2 Сильно выпуклая постановка	13
5 Вычислительный эксперимент	15
5.1 Результаты на CIFAR-10 с использованием ResNet-18	15
5.2 Результаты на CIFAR-100 с использованием ResNet-18	16
5.3 Результаты на Tiny ImageNet с использованием Swin Transformer	17
Список литературы	19
Приложение	22

1 Введение

В последние годы в области машинного обучения наблюдается значительный прогресс, обусловленный стремлением к повышению качества решений и возможности решать всё более сложные задачи. Это привело к заметному увеличению как объёма данных, так и масштабов моделей. Данные изменения являются критически важными, поскольку способствуют стабилизации результатов и повышению точности выполнения задач.

Большинство задач машинного обучения сводится к задаче минимизации суммы конечного числа функций:

$$\min_{x \in \mathbb{R}^d} \left[f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \right],$$

где $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$, а число функций n велико. Например, в задаче обучения моделей машинного обучения, n соответствует размеру обучающей выборки, а $f_i(x)$ — функции потерь модели на i -м объекте, где x — вектор параметров модели.

Стохастические методы хорошо подходят для данной задачи, так как позволяют избежать вычисления полного градиента на каждой итерации. В условиях реальных задач, где n может быть чрезвычайно велико, такие вычисления становятся крайне ресурсоёмкими. Одним из наиболее известных методов является стохастический градиентный спуск (SGD) и его модификации. На t -й итерации метод выбирает индекс $i_t \in \{1, \dots, n\}$ и выполняет шаг:

$$x^{t+1} = x^t - \gamma \nabla f_{i_t}(x^t),$$

где γ — величина шага метода.

Методы уменьшения дисперсии

Несмотря на простоту SGD, он обладает существенным недостатком: дисперсия оценок градиента сохраняется большой на протяжении всего обучения.

В результате, при использовании постоянного шага, метод сходится лишь к окрестности оптимального решения, размер которой зависит от дисперсии. Для решения проблемы высокой дисперсии стохастических градиентных методов были предложены методы уменьшения дисперсии (Variance Reduction, VR), такие как SAG [1], SAGA [2], FINITO [3], SPIDER [4] и SARAH [5]. Эти методы основаны на идее построения более точных оценок градиента, что позволяет существенно улучшить скорость сходимости по сравнению с классическим SGD [6]. В частности, метод SARAH использует рекурсивное обновление оценок градиента и обладает как теоретическими, так и практическими преимуществами при оптимизации больших моделей.

Алгоритм 1 SARAH: StochAstic Recursive grAdient algoritHm

Require: начальная точка x^0 , шаг γ , период обновления m

```

1: for  $s = 0, 1, 2, \dots$  do
2:    $v^0 = \nabla f(x^0)$ 
3:    $x^1 = x^0 - \gamma v^0$ 
4:   for  $t = 1$  to  $m$  do
5:     выбрать  $i_t \sim \text{Uniform}(\{1, \dots, n\})$ 
6:      $v^t = \nabla f_{i_t}(x^t) - \nabla f_{i_t}(x^{t-1}) + v^{t-1}$ 
7:      $x^{t+1} = x^t - \gamma v^t$ 
8:   end for
9:    $x^0 \leftarrow x^{m+1}$                                  $\triangleright$  Перезапуск
10: end for

```

В данной работе анализу подвергается метод SARAH, в котором оценка градиента обновляется рекурсивно:

$$v^t = \nabla f_{i_t}(x^t) - \nabla f_{i_t}(x^{t-1}) + v^{t-1}; \quad x^{t+1} = x^t - \gamma v^t,$$

где γ — величина шага оптимизации. Для достижения сходимости к оптимальному решению x^* требуется периодически пересчитывать v^t с использованием

полного градиента. Эта процедура выполняется либо через фиксированное число итераций, либо случайным образом. Практическая версия метода, известная как SARAH+, использует правило автоматического выбора момента обновления, основываясь на величине отношения $\|v^t\|/\|v^0\|$.

Эвристика перемешивания

Важным, но часто недооцениваемым аспектом стохастических методов является способ выбора индексов на каждой итерации. Это напрямую влияет как на стабильность, так и на сходимость. Вместо случайного выбора на каждой итерации, в настоящей работе применяется эвристика перемешивания. Изначально создаётся случайная перестановка индексов $\{1, \dots, n\}$, после чего на каждой итерации используется соответствующий элемент этой перестановки.

Наиболее известные варианты включают:

- **Random Reshuffle (RR)** — данные перемешиваются перед каждой эпохой;
- **Shuffle Once (SO)** — перемешивание только один раз в начале обучения;
- **Cyclic** — данные проходят в фиксированном порядке без перемешивания.

Во всех вариантах перемешивания градиент для каждого объекта вычисляется ровно один раз за эпоху. При этом, выбор по перестановке нарушает свойство несмещённости градиентных оценок:

$$\mathbb{E}_{\pi_s^t} [\nabla f_{\pi_s^t}(x_s^t)] \neq \nabla f(x_s^t),$$

что приводит к более сложному анализу и нестандартным техникам доказательства.

2 Обзор литературы

Методы без вычисления полного градиента

Метод SARAH на сегодняшний день является одним из стандартных подходов к решению задачи минимизации суммы. Однако, классическая версия требует периодического вычисления полного градиента. В связи с этим был проявлен интерес к вариантам, избегающим таких вычислений.

Методы SAG и SAGA решают эту задачу, но требуют хранения дополнительных градиентов, что ведёт к затратам памяти порядка $\mathcal{O}(nd)$. Ряд подходов был предложен для модификации SARAH, исключающей необходимость в полном градиенте.

- В работе [7] предложен алгоритм inexact-SARAH, в котором полный градиент заменяется на минибатч-оценку:

$$\frac{1}{|S|} \sum_{i \in S} f_i(x), \quad S \subset \{1, \dots, n\}.$$

- В других работах предлагается гибридная схема без рестартов:

$$v^t = \beta_t \nabla f_{i_t}(x^t) + (1 - \beta_t)(\nabla f_{i_t}(x^t) - \nabla f_{i_t}(x^{t-1}) + v^{t-1}),$$

где параметр β_t либо постоянен [8], либо стремится к нулю, как в методе STORM [9]. Метод ZEROSARAH [10] сочетает такую схему с SAG/SAGA.

Методы с перемешиванием

Так как рассматриваемый метод использует эвристику перемешивания, необходимо проанализировать существующие подходы. При выборке без возвращения на протяжении эпохи каждый объект используется ровно один раз. Было показано, что Random Reshuffle может сходиться быстрее, чем SGD, на практике.

Однако теоретические оценки долгое время отставали от классических методов с независимым выбором индексов. Прорыв был достигнут в работе [11], где представлены новые методы анализа. Для сильно выпуклых задач получены такие же оценки, как у SGD с независимым выбором. Однако в невыпуклом случае результаты остались слабее, а также требовали большого числа эпох, что не характерно для современных нейросетей. Альтернатива была предложена в [12], где анализ строится на так называемом «периоде корреляции» вместо полной эпохи.

Позднее эвристика перемешивания была применена к более общим задачам вариационных неравенств и, в частности, к методу EXTRAGRADIENT. Это позволило получить аналогичные линейные оценки. В дальнейшем внимание было сосредоточено на сочетании перемешивания с методами уменьшения дисперсии.

Таблица 1: Сравнение оценок сходимости различных алгоритмов

Алгоритм	Без полного градиента?	Память	Невыпуклая задача	Сильно выпуклая задача
SAGA [13]	✓	$\mathcal{O}(nd)$	\	$\mathcal{O}\left(n \frac{L^2}{\mu^2} \log\left(\frac{1}{\varepsilon}\right)\right)$
IAG [14]	✓	$\mathcal{O}(nd)$	\	$\mathcal{O}\left(n^2 \frac{L^2}{\mu^2} \log\left(\frac{1}{\varepsilon}\right)\right)$
PIAG [15]	✓	$\mathcal{O}(nd)$	\	$\mathcal{O}\left(n \frac{L}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$
DIAG [16]	✓	$\mathcal{O}(nd)$	\	$\mathcal{O}\left(n \frac{L}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$
Prox-DFinito [17]	✓	$\mathcal{O}(nd)$	\	$\mathcal{O}\left(n \frac{L}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$
AVRG [18]	✓	$\mathcal{O}(d)$	\	$\mathcal{O}\left(n \frac{L^2}{\mu^2} \log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG [19]	✗	$\mathcal{O}(d)$	\	$\mathcal{O}\left(n^3 \frac{L^2}{\mu^2} \log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG [20]	✗	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n \frac{L^{3/2}}{\mu^{3/2}} \log\left(\frac{1}{\varepsilon}\right)\right)^{(1)}$
SARAH [21]	✓	$\mathcal{O}(d)$	\	$\mathcal{O}\left(n^2 \frac{L}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$
SARAH (данная работа)	✓	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n \frac{L}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$

Столбцы: «Без полного градиента?» — указывает, требует ли метод вычисления полного градиента. «Память» — дополнительные затраты по памяти.

Обозначения: μ — коэффициент сильной выпуклости, L — константа гладкости, n — размер выборки, d — размерность задачи, ε — требуемая точность.

(1) В данной работе также получены улучшенные оценки в случае больших данных: $n \gg \mathcal{O}\left(\frac{L}{\mu}\right)$, однако они выходят за рамки настоящего анализа.

3 Постановка задачи

В данной работе рассматривается задача оптимизации конечной суммы функций следующего вида:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x),$$

где каждая функция $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ соответствует функции потерь на одном элементе выборки. Предполагается, что f дифференцируема, однако может быть как выпуклой, так и невыпуклой. Основная цель — нахождение точки, минимизирующей функцию f с использованием стохастических градиентных методов, не требующих вычисления полного градиента на каждой итерации.

Для теоретического анализа вводится ряд стандартных предположений.

Предположение 1 (Гладкость функций). *Каждая функция f_i обладает L -гладкостью, то есть для любых $x, y \in \mathbb{R}^d$ выполняется неравенство:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|.$$

Предположение 2 (Сильная выпуклость). *Каждая функция f_i является μ -сильно выпуклой, то есть для любых $x, y \in \mathbb{R}^d$ выполняется:*

$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Предположение 3 (Невыпуклость). *Функция f может быть невыпуклой, но при этом обладает конечным инфимумом:*

$$f^* := \inf_{x \in \mathbb{R}^d} f(x) > -\infty.$$

4 Основные результаты

4.1 Аппроксимация полного градиента

В данном разделе представлена эвристика, основанная на перемешивании выборки и идеях SAG/SAGA, позволяющая аппроксимировать полный градиент без необходимости хранения всех предыдущих значений градиентов. Алгоритм SAG был одним из первых методов, направленных на ускорение стохастического градиентного спуска за счёт уменьшения дисперсии обновлений. В SAG обновление имеет вид:

$$x^{t+1} = x^t - \frac{\gamma}{n} \left(\nabla f_{i_t}(x^t) - \nabla f_{i_t}(\phi_{i_t}^t) + \sum_{j=1}^n \nabla f_j(\phi_j^t) \right), \quad (1)$$

где ϕ_j^t — точка, в которой ранее был вычислен градиент функции f_j . В данном подходе на каждом шаге обновляется один из элементов суммы, что снижает дисперсию оценки градиента.

При случайной выборке i_t в SAG сложно отследить, когда последний раз обновлялся градиент для конкретного индекса. Однако при использовании перемешивания (shuffling) известно, что в течение эпохи все градиенты ∇f_j будут вычислены. Таким образом, в начале каждой эпохи возможна точная аппроксимация полного градиента:

$$v_{s+1} = \frac{1}{n} \sum_{t=1}^n \nabla f_{\pi_s^t}(x_s^t), \quad (2)$$

где π_s^t — перестановка индексов после перемешивания в начале эпохи s . При этом расчёт может быть реализован через скользящее среднее без дополнительных затрат памяти:

$$\tilde{v}_s^{t+1} = \frac{t}{t+1} \tilde{v}_s^t + \frac{1}{t+1} \nabla f_{\pi_s^t}(x_s^t), \quad v_{s+1} = \tilde{v}_s^n. \quad (3)$$

Корректность формулы (3) относительно (2) показывается в доказательстве леммы ??.

4.2 SARAH без полного градиента

Алгоритм SARAH зарекомендовал себя как эффективный метод снижения дисперсии градиентных оценок, обладающий практическими преимуществами по сравнению с альтернативами. В данном разделе рассматривается модификация алгоритма, исключающая необходимость пересчёта полного градиента. Ниже представлена формальная постановка алгоритма.

Алгоритм 2 No FULL GRAD SARAH

- 1: **Вход:** Начальное приближение $x_0^0 \in \mathbb{R}^d$; Начальные градиенты $\tilde{v}_0^0 = 0^d, v_0 = 0^d$
 - 2: **Параметр:** Шаг градиентного спуска $\gamma > 0$
 - 3: **for** эпохи $s = 0, 1, 2, \dots, S$ **do**
 - 4: Сэмплируется перестановка π_s^1, \dots, π_s^n из $\overline{1, n}$ ▷ по эвристике
 - 5: $v_s^0 = v_s$
 - 6: $x_s^1 = x_s^0 - \gamma v_s^0$
 - 7: **for** $t = 1, 2, \dots, n$ **do**
 - 8: $\tilde{v}_s^{t+1} = \frac{t-1}{t}\tilde{v}_s^t + \frac{1}{t}\nabla f_{\pi_s^t}(x_s^t)$
 - 9: $v_s^t = \frac{1}{n}(\nabla f_{\pi_s^t}(x_s^t) - \nabla f_{\pi_s^t}(x_s^{t-1})) + v_s^{t-1}$
 - 10: $x_s^{t+1} = x_s^t - \gamma v_s^t$
 - 11: **end for**
 - 12: $x_{s+1}^0 = x_s^{n+1}$
 - 13: $\tilde{v}_{s+1}^1 = 0$
 - 14: $v_{s+1} = \tilde{v}_s^{n+1}$
 - 15: **end for**
-

Модификация алгоритма позволяет отказаться от пересчёта полного градиента, используя идею скользящего усреднения стохастических градиентов. Обновление в строке 8 учитывает изменение индексации: усреднение начинает-

ся с $t = 1$, а не с $t = 0$, что позволяет избежать появления лишнего множителя $\frac{1}{n+1}$ в оценках.

4.2.1 Невыпуклая постановка

Анализ проводится аналогично предыдущему разделу. Вначале доказывается оценка изменения градиента на одной эпохе:

Лемма 1. *Пусть выполнены предположения 1, 2, 3. Тогда для алгоритма 2 справедлива оценка:*

$$\left\| \nabla f(x_s^0) - \frac{1}{n+1} \sum_{t=0}^n v_s^t \right\|^2 \leq 2\|\nabla f(x_s^0) - v_s\|^2 + \frac{2L^2}{n+1} \sum_{t=1}^n \|x_s^t - x_s^{t-1}\|^2.$$

Полученное выражение содержит два слагаемых: первое отвечает за точность аппроксимации полного градиента, второе — за накопленную ошибку, возникающую при отклонении траектории оптимизации от начальной точки. Далее доказывается следующая лемма:

Лемма 2. *Пусть выполнены предположения 1, 2, 3 и шаг $\gamma \leq \frac{1}{3L}$. Тогда для алгоритма 2 справедливо:*

$$\left\| \nabla f(x_s^0) - \frac{1}{n+1} \sum_{t=0}^n v_s^t \right\|^2 \leq 9\gamma^2 L^2 \|v_s\|^2 + 36\gamma^2 L^2 n^2 \|v_{s-1}\|^2.$$

На основании этих лемм формулируется основная теорема:

Теорема 1. *Пусть выполнены предположения 1, 3. Тогда алгоритм 2 при шаге $\gamma \leq \frac{1}{20L(n+1)}$ достигает ε -точности, определяемой как $\varepsilon^2 = \frac{1}{S} \sum_{s=1}^S \|\nabla f(x_s^0)\|^2$, за $\mathcal{O}(nL/\varepsilon^2)$ итераций и вызовов оракула.*

4.2.2 Сильно выпуклая постановка

Переход к сильно выпуклой постановке осуществляется с использованием условия Поляка-Ложасье (см. Приложение А).

Теорема 2. Пусть выполнены предположения 1, 2. Тогда алгоритм 2 при шаге $\gamma \leq \frac{1}{20L(n+1)}$ достигает ε -точности, определяемой как $\varepsilon = f(x_{S+1}^0) - f(x^*)$, за $\mathcal{O}(nL/\mu \log^{1/\varepsilon})$ итераций и вызовов оракула.

5 Вычислительный эксперимент

Целью данного раздела является эмпирическая проверка эффективности предложенного алгоритма NO FULL GRAD SARAH на задачах классификации изображений. Были проведены вычислительные эксперименты на следующих датасетах:

- CIFAR-10 и CIFAR-100 с использованием архитектуры ResNet-18;
- TINY IMAGENET с использованием модели Swin Transformer.

Во всех экспериментах обучение осуществлялось с размером батча 128. Параметр регуляризации веса был установлен как $\lambda_1 = 5 \times 10^{-4}$. Для каждой модели фиксировались метрики качества на обучающей и тестовой выборках: кросс-энтропийная функция потерь и точность. Метрики визуализировались в зависимости от числа эквивалентных вызовов полного градиента.

5.1 Результаты на CIFAR-10 с использованием ResNet-18

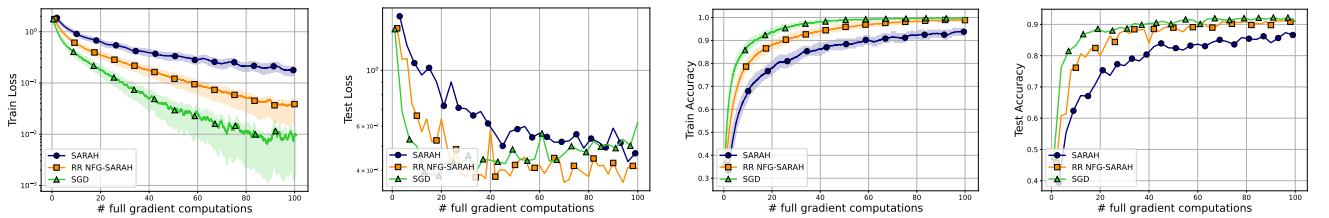


Рис. 1: Сходимость NO FULL GRAD SARAH и классического SARAH на датасете CIFAR-10.

Полученные результаты демонстрируют стабильное убывание функции потерь при обучении методом NO FULL GRAD SARAH, опережая оригинальный алгоритм по скорости сходимости при одинаковом количестве эквивалентных вызовов полного градиента. На тестовой выборке также наблюдается

уменьшение функции потерь до более низкого значения, при этом финальная точность постепенно улучшается и достигает более высоких значений на поздних стадиях обучения.

5.2 Результаты на CIFAR-100 с использованием ResNet-18

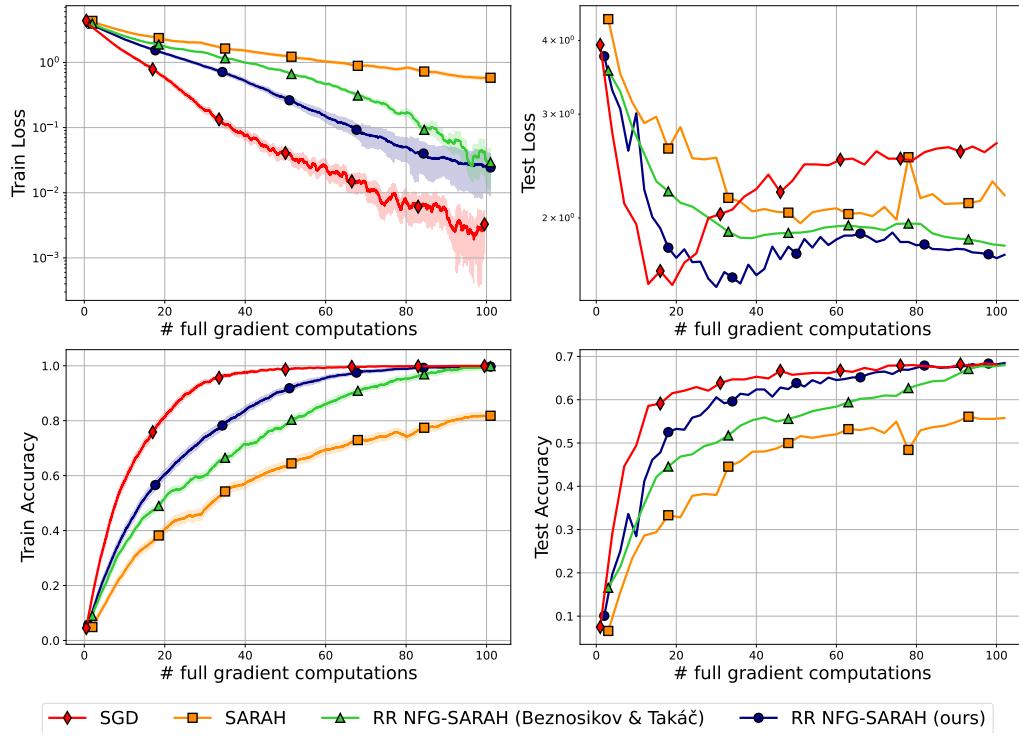


Рис. 2: Сходимость NO FULL GRAD SARAH и SARAH на датасете CIFAR-100.

На CIFAR-100 также была получена устойчивая сходимость предложенного метода. Тестовая ошибка продолжает уменьшаться, даже после того как классический метод достигает минимума. Отметим также, что точность возрастает значительно быстрее на поздних стадиях обучения, что указывает на улучшенное обобщающее поведение.

5.3 Результаты на Tiny ImageNet с использованием Swin Transformer

Был использован датасет Tiny ImageNet, включающий 200 классов изображений размером 64×64 , увеличенных до 224×224 с целью соответствия входу модели. Архитектура — Tiny Swin Transformer (`swin_T_patch4_window7_224`), инициализированная предобученными весами с ImageNet-1K. Обучение велось с градиентным клиппингом на уровне 1.0. Размер батча — 256.

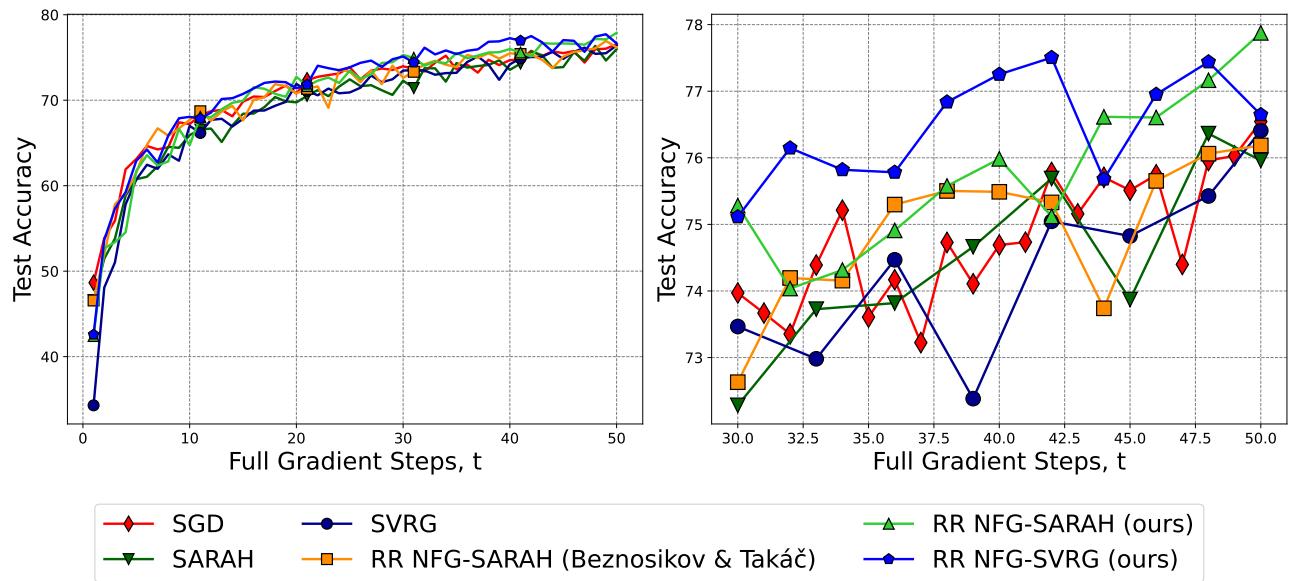


Рис. 3: Сходимость NO FULL GRAD SARAH и других методов на Tiny ImageNet.

Таблица 2: Финальная точность различных методов на Tiny ImageNet.

Метод	Точность (\uparrow)
SGD	76.545
SARAH	75.961
RR NFG-SARAH (по [21])	76.186
RR NFG-SARAH (предложенный)	77.875

Модифицированный алгоритм NO FULL GRAD SARAH, реализованный в рамках данной работы, демонстрирует превосходство над как классическими, так и ранее предложенными модифицированными алгоритмами. Особенно заметно улучшение в задачах с большим числом параметров и высокой сложностью модели, таких как Swin Transformer.

Список литературы

- [1] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112, 2017.
- [2] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.
- [3] Aaron Defazio, Justin Domke, et al. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning*, pages 1125–1133. PMLR, 2014.
- [4] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.
- [5] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.
- [6] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [7] Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021.
- [8] Deyi Liu, Lam M Nguyen, and Quoc Tran-Dinh. An optimal hybrid variance-reduced algorithm for stochastic composite nonconvex optimization. *arXiv preprint arXiv:2008.09055*, 2020.
- [9] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction

in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

- [10] Zhize Li, Slavomír Hanzely, and Peter Richtárik. Zerosarah: Efficient nonconvex finite-sum optimization with zero full gradient computation. *arXiv preprint arXiv:2103.01447*, 2021.
- [11] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.
- [12] Anastasia Koloskova, Nikita Doikov, Sebastian U. Stich, and Martin Jaggi. On convergence of incremental gradient for non-convex smooth functions, 2024.
- [13] Youngsuk Park and Ernest K Ryu. Linear convergence of cyclic saga. *Optimization Letters*, 14(6):1583–1598, 2020.
- [14] Mert Gurbuzbalaban, Asuman Ozdaglar, and Pablo A Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.
- [15] Nuri Denizcan Vanli, Mert Gurbuzbalaban, and Asu Ozdaglar. A stronger convergence result on the proximal incremental aggregated gradient method. *arXiv preprint arXiv:1611.08022*, 2016.
- [16] Aryan Mokhtari, Mert Gurbuzbalaban, and Alejandro Ribeiro. Surpassing gradient descent provably: A cyclic incremental method with linear convergence rate. *SIAM Journal on Optimization*, 28(2):1420–1447, 2018.
- [17] Xinpeng Huang, Kun Yuan, Xianghui Mao, and Wotao Yin. An improved analysis and rates for variance reduction under without-replacement sampling orders. *Advances in Neural Information Processing Systems*, 34:3232–3243, 2021.

- [18] Bicheng Ying, Kun Yuan, and Ali H Sayed. Variance-reduced stochastic learning under random reshuffling. *IEEE Transactions on Signal Processing*, 68:1390–1408, 2020.
- [19] Tao Sun, Yuejiao Sun, Dongsheng Li, and Qing Liao. General proximal incremental aggregated gradient algorithms: Better and novel results under general scheme. *Advances in Neural Information Processing Systems*, 32, 2019.
- [20] Grigory Malinovsky, Alibek Sailanbayev, and Peter Richtárik. Random reshuffling with variance reduction: New analysis and better rates. In *Uncertainty in Artificial Intelligence*, pages 1347–1357. PMLR, 2023.
- [21] Aleksandr Beznosikov and Martin Takáč. Random-reshuffled sarah does not need full gradient computations. *Optimization Letters*, pages 1–23, 2023.
- [22] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Приложение

A Основные неравенства

В этом разделе приводятся неравенства, используемые в дальнейшем. Пусть функция f удовлетворяет Предположению 1, а функция g — Предположению 2. Тогда для любых векторов $x, y, \{x_i\} \in \mathbb{R}^d$ и положительных скаляров α, β выполняются следующие неравенства:

$$2\langle x, y \rangle \leq \frac{\|x\|^2}{\alpha} + \alpha \|y\|^2, \quad (\text{Скалярное})$$

$$2\langle x, y \rangle = \|x + y\|^2 - \|x\|^2 - \|y\|^2, \quad (\text{Норма})$$

$$\|x + y\|^2 \leq (1 + \beta)\|x\|^2 + \left(1 + \frac{1}{\beta}\right)\|y\|^2, \quad (\text{Квадратичное})$$

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2, \quad (\text{Липшицево})$$

$$\left\| \sum_{i=1}^n x_i \right\|^2 \leq n \sum_{i=1}^n \|x_i\|^2, \quad (\text{Коши--Буняковский})$$

$$g(x) - \inf g \leq \frac{1}{2\mu} \|\nabla g(x)\|^2. \quad (\text{PL-условие})$$

Неравенство (Липшицево) получено в книге [22], Теорема 2.1.5.