Методы редукции дисперсии, не предполагающие вычисление полного градиента: повышение эффективности за счёт техники случайного перемешивания батчей

Алексей Витальевич Ребриков
Научный руководитель: к.ф.-м.н. А. Н. Безносиков

Кафедра интеллектуальных систем ФПМИ МФТИ
Специализация: Интеллектуальный анализ данных
Направление: 03.03.01 Прикладные математика и физика

2025

Редукция дисперсии и полный градиент

Проблема: оптимизация конечной суммы:

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

- GD: вычисляет полный градиент $\Rightarrow$ долго при большом n.
- SGD: не требует полного градиента $\Rightarrow$ дисперсия решения.
- Классические методы редукции дисперсии (VR, Variance Reduction) периодически вычисляют полный градиент.

Редукция дисперсии и полный градиент

Проблема: оптимизация конечной суммы:

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

- GD: вычисляет полный градиент $\Rightarrow$ долго при большом n.
- SGD: не требует полного градиента $\Rightarrow$ дисперсия решения.
- Классические методы редукции дисперсии (VR, Variance Reduction) периодически вычисляют полный градиент.

Замечание

Стохастику ввели, чтобы избежать полного градиента, но в VR-методах к нему вернулись (хоть и реже) — замкнутый круг.

Редукция дисперсии и полный градиент

Проблема: оптимизация конечной суммы:

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

- GD: вычисляет полный градиент $\Rightarrow$ долго при большом n.
- SGD: не требует полного градиента $\Rightarrow$ дисперсия решения.
- Классические методы редукции дисперсии (VR, Variance Reduction) периодически вычисляют полный градиент.

Цель: Метод VR без вычисления полного градиента.

Решение: Модификация алгоритма `SARAH`.
Лучшие оценки сходимости — за счёт перемешивания батчей.

SARAH (StochAstic Recursive grAdient algoritHm)

Метод стохастической оптимизации с редукцией дисперсии.

В отличие от `SGD` **рекурсивное обновление градиента**:

- уменьшает дисперсию оценок градиента без хранения всех,
- обходится редкими обращениями к полному градиенту.

Алгоритм:

- *Начало эпохи:* $v^0 = \nabla f(x^0)$ — полный градиент
- *Внутри эпохи:*

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

Lam M Nguyen et al. (2017). "SARAH: A novel method for machine learning problems using stochastic recursive gradient". In: *International conference on machine learning*. PMLR, с. 2613—2621

Модификация SARAH: RR No Full Grad SARAH

Алгоритм:

- *Начало эпохи:* $v^0 = \tilde{\mathbf{v}}^{\mathbf{n}}$ или начальная оценка (например, 0)
  где $\tilde{\mathbf{v}}^{\mathbf{t}}$ — скользящая оценка полного градиента на шаге t
- *Внутри эпохи:*

$$v^t = \frac{\mathbf{1}}{\mathbf{n}} \left(\nabla f_{i_t}(x^t) - \nabla f_{i_t}(x^{t-1}) \right) + v^{t-1}, \quad x^{t+1} = x^t - \gamma v^t$$

$$\tilde{\mathbf{v}}^{\mathbf{t+1}} = \frac{t-1}{t} \cdot \tilde{\mathbf{v}}^{\mathbf{t}} + \frac{1}{t} \cdot \nabla f_{i_t}(x^t)$$

Особенности:

- Перемешивание батчей — каждый ровно один раз за эпоху.
- Вместо полного градиента v^0 **скользящее среднее** $\tilde{\mathbf{v}}$.
- **Домножение на** $1/n \Rightarrow$ дополнительно VR.

Первая идея исследовалась в статье:
Aleksandr Beznosikov and Martin Takáč (2023). "Random-reshuffled SARAH does not need full gradient computations". In: *Optimization Letters*, с. 1—23

Теоретические результаты

Все f_i — L-гладкие, шаг $\gamma \leq \frac{1}{20L(n+1)}$, n — размер выборки.

Невыпуклый случай:

$$\varepsilon^2 = \frac{1}{S}\sum_{s=1}^{S} \|\nabla f(x_s^0)\|^2 \qquad \mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$$

Сильно выпуклый случай:

$$\varepsilon = f(x_{S+1}^0) - f(x^*) \qquad \mathcal{O}\left(\frac{nL}{\mu}\log\frac{1}{\varepsilon}\right)$$

Предложенный алгоритм лучше красного

Алгоритм	**Нет полного градиента?**	**Память**	**Невып.**	**Сильно вып.**
SAGA	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
IAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n^2\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
PIAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
DIAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
Prox-DFinito	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
AVRG	✓	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
SVRG	✗	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n^3\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
SVRG	✗	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n\frac{L^{3/2}}{\mu^{3/2}}\log\frac{1}{\varepsilon}\right)$
SARAH	✓	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n^2\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
RR NFG SARAH	✓	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$

Эксперимент: CIFAR-10 + ResNet18

Многоклассовая классификация на CIFAR-10:

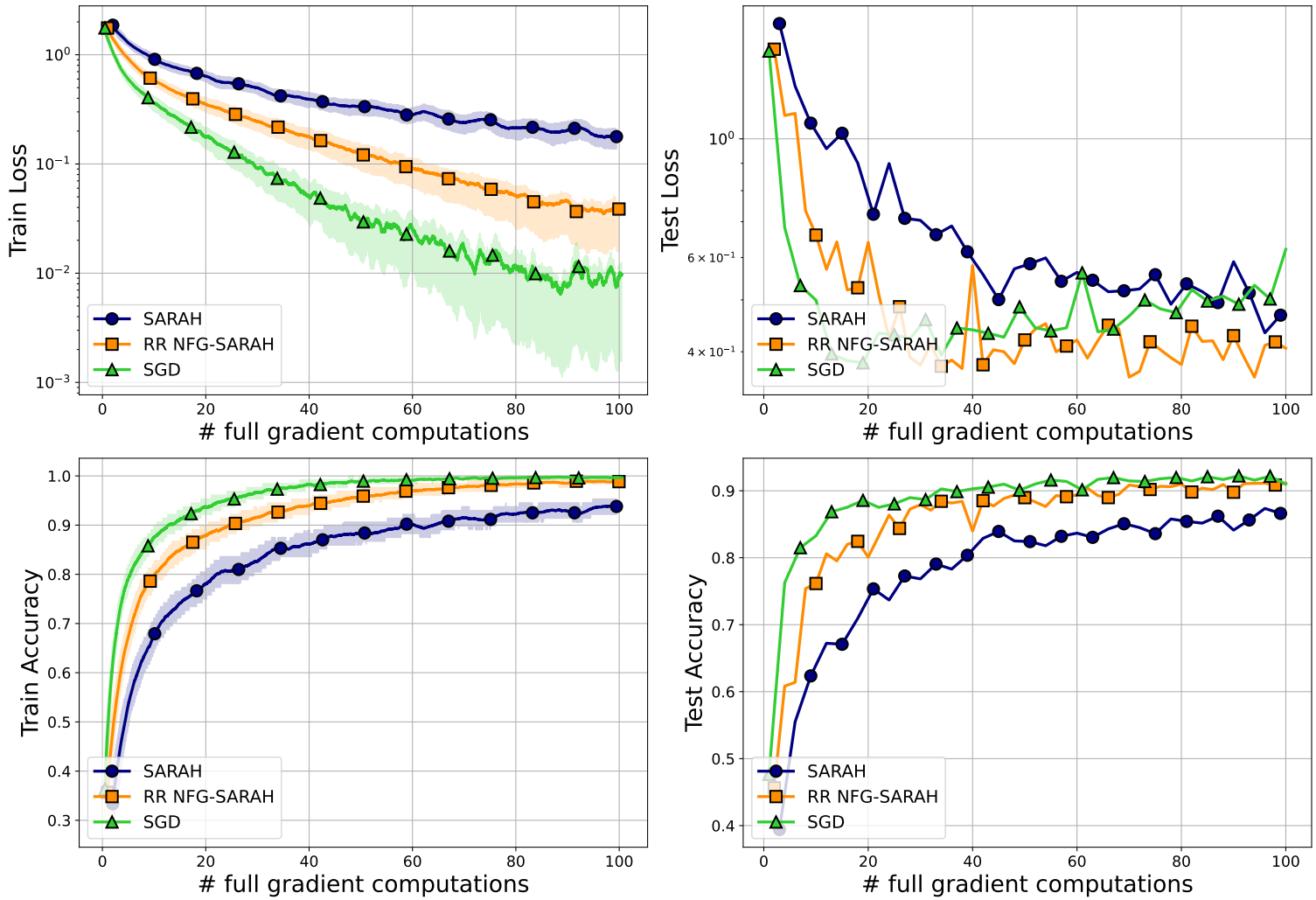
- 60 000 изображений 32×32 в 10 классах
- Архитектура: ResNet-18
- Функция потерь: кросс-энтропия

$$\min_w \frac{1}{M} \sum_{i=1}^{M} \ell(f_w(x_i), y_i)$$

- Метрики: точность и кросс-энтропия
- Все методы сравниваются по числу эквивалентных вызовов полного градиента

Подробности эксперимента на CIFAR-100 см. в работе.

Сходимость на CIFAR-10 (ResNet-18)



Везде сравнимо или лучше, чем `SGD`. Везде лучше, чем `SARAH`.

Эксперимент: Swin Transformer + Tiny ImageNet

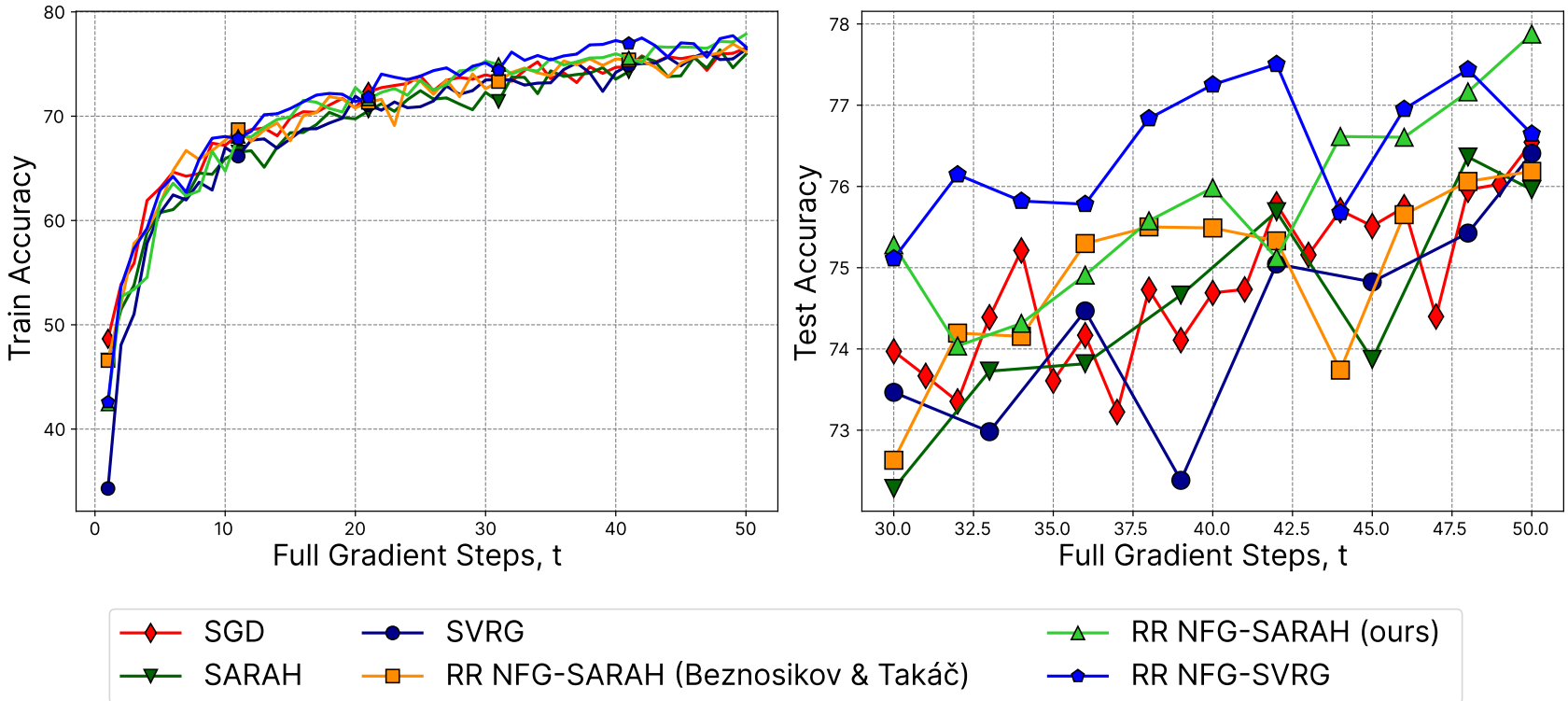
Многоклассовая классификация изображений:

- Tiny ImageNet: 200 классов, 64×64, масштаб до 224×224
- Модель: Tiny Swin Transformer (`swin_T_patch4_window7_224`) с предобучением на ImageNet-1K
- Функция потерь: кросс-энтропия

$$\min_w \frac{1}{M} \sum_{i=1}^{M} \ell(f_w(x_i), y_i)$$

- Метрики: точность и кросс-энтропия
- Все методы сравниваются по числу эквивалентных вызовов полного градиента

Сходимость на Tiny ImageNet (Swin Transformer)



Предложенный метод достигает лучшей точности. Сходимость на тестовой выборке стабильнее остальных.

Дополнительно приведен алгоритм из статьи (упоминалась на 4 слайде):
Aleksandr Beznosikov and Martin Takáč (2023). "Random-reshuffled SARAH does not need full gradient computations". In: *Optimization Letters*, с. 1—23

Выносится на защиту

- Разработан новый вариант метода **SARAH**, не использующий вычисление полного градиента.
- За счёт перемешивания и скользящего среднего достигается аппроксимация полного градиента без дополнительного хранения.
  - **Память:** требуется $\mathcal{O}(d)$ вместо $\mathcal{O}(nd)$
  - **Сходимость:** улучшенные оценки по числу итераций
- Проведена серия экспериментов (CIFAR-10/CIFAR-100 с ResNet-18, Tiny ImageNet с Swin Transformer), подтверждающих теоретические результаты.
- Работа была представлена на 67-й Всероссийской научной конференции МФТИ.
- Полученные результаты объединены с другими исследованиями и поданы в виде статьи на международную конференцию уровня A* в области машинного обучения.