ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2024, ТОМ 64, № 4, с. 575–586

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

УДК 517.97

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК С ПРЕДОБУСЛОВЛЕННЫМ РАЗМЕРОМ ШАГА им. Б. Т. ПОЛЯКА

© 2024 г. Ф. Абдухакимов[1],*, Ч. Сян[1],, Д. Камзолов[1],***, М. Такач[1],****

[1] *Университет искусственного интеллекта им. Мохамеда бин Заеда, Абу-Даби, ОАЭ*

** e-mail: farshed888@gmail.com*

*** e-mail: chulu.xiang@mbzuai.ac.ae*

**** e-mail: kamzolov.opt@gmail.com*

***** e-mail: takac.mt@gmail.com*

Поступила в редакцию 02.11.2023 г.
Переработанный вариант 16.12.2023 г.
Принята к публикации 20.12.2023 г.

Стохастический градиентный спуск (SGD) является одним из множества методов оптимизации, используемых для решения задач машинного обучения. Практичность и простота подобных методов привлекают не только исследователей, но и инженеров машинного обучения из индустрии. Однако одна из главных слабостей таких методов заключается в необходимости ручной настройки размера шага для эффективного решения каждой конкретной оптимизационной задачи, функции потерь и данных. Стохастический градиентный спуск с размером шага им. Б.Т. Поляка (SPS) – это метод, который предлагает правило обновления, не требующее точной ручной настройки размера шага для решения задачи. Цель настоящей работы – расширить SPS с помощью таких приемов предобуславливания, как методы Хатчинсона, Adam и AdaGrad, что, в свою очередь, улучшит эффективность SPS в случае с плохой обусловленностью задачи и данных. Библ. 31. Фиг. 5.

Ключевые слова: машинное обучение, оптимизация, адаптивный размер шага, размер шага им. Б.Т. Поляка, предобусловленность.

DOI: 10.31857/S0044466924040016, **EDN:** ZKLWGL

1. ВВЕДЕНИЕ

В настоящей статье мы рассматриваем задачу минимизации эмпирического риска (МЭР, англ. Empirical risk minimization, ERM), имеющую вид оптимизации конечной суммы:

$$w^* \in \operatorname*{argmin}_{w\in\mathbb{R}^d}\left\{f(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w)\right\}, \tag{1}$$

где $w \in \mathbb{R}^d$ является параметром весов и каждая целевая функция $f_i : \mathbb{R}^d \to \mathbb{R}$ является гладкой и дважды дифференцируемой. Функция потерь $f_i(w)$ вычисляет разницу между предсказанием модели с параметрами весов w и целевым значением y. Целью является минимизация средней потери $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$ на n данных $\{(x_i, y_i)\}_{i=1}^{n}$, где x_i — входная точка данных и y_i — соответствующее целевое значение. В связи с нетривиальностью данной задачи решение в явном виде не всегда доступно, что подталкивает к применению численных методов оптимизации. Одним из таких численных методов является стохастический градиентный спуск (SGD) со следующим обновлением параметров весов:

$$w_{t+1} = w_t - \gamma_t \nabla f_i(w_t), \tag{2}$$

где $\gamma_t \in \mathbb{R}$ — размер шага метода. Использование мини-батчей для датасетов с большой размерностью при обучении значительно уменьшает время сходимости к оптимальной точке w^*. Были проведены обширные исследования в области стохастических методов оптимизации первого порядка начиная с фундаментальных работ Г. Роббинса и С. Монро [1], Б. Т. Поляка [2], Б. Т. Поляка и А. Б. Юдицкого [3], А. С. Немировского и др. [4] и ускоренные версии от Г. Лан [5]. Стоить отметить, что каждая комбинация функции потерь и датасета требует отдельной ручной настройки размера шага γ_t для поиска минимума, что делает γ_t гиперпараметром. Эта проблема ручной настройки γ_t является одним из мотивирующим факторов

разработки методов с адаптивным размером шага, где γ_t заменена адаптивно меняющимся выражением по ходу оптимизации. В последнее время такие адаптивные методы получили широкое распространение (см. [6]–[13]) особенно в области обучения глубоких нейронных сетей.

Другое направление адаптивных стохастических методов — стохастический градиентный спуск с размером шага им. Б.Т. Поляка, который был вдохновлен размером шага для субградиентных методов, предложенным Борисом Теодоровичем Поляком в 1969 г. (см. [14], [15]). Позже был предложен стохастический вариант этого шага в [16], [17] и другие различные расширения в [18]–[24]. В следующем разделе мы детально разберем некоторые из них.

Одним из главных предметов обсуждения статьи является получение методов, предназначенных для решения задач с плохой обусловленностью с помощью техники предобуславливания градиента. Несмотря на то что достижение идеального предобусловливания практически невозможно, наше решение использует различные техники, предложенные в таких адаптивных алгоритмах, как Adam [7] и AdaGrad [6], а также метод Хатчинсона [25].

Введем обозначения. Мы наделяем прямое пространство $w \in \mathbf{E}$ и двойственное пространство $g \in \mathbf{E}^*$ сопряженными нормами $\|w\|$ и $\|g\|_*$ соответственно. Как частный случай для положительно-определенной матрицы $B \in \mathbb{R}^{d\times d}$ мы определяем двойственные евклидовы нормы: $\|w\|_B = \langle Bw, w\rangle^{1/2}$ и $\|g\|_{B^{-1}} = \left\langle g, B^{-1}g\right\rangle^{1/2}$. Отметим, что $\nabla f(w) \in \mathbf{E}^*$ и $\nabla^2 f(w)h \in \mathbf{E}^*$ для $h \in \mathbf{E}$. Оператор $\odot$ определяется как покомпонентное умножение двух векторов, также известное как произведение Адамара. Мы обозначаем $\mathrm{diag}(v)$ диагональную матрицу по заданному вектору v и вектор $\mathrm{diagonal}(\mathbf{H}) \in \mathbb{R}^d$ как диагональ матрицы $\mathbf{H} \in \mathbb{R}^{d\times d}$. Для простоты мы также вводим следующее обозначение: $(x)_+ = \max\{0, x\}$.

2. ОБЗОР ЛИТЕРАТУРЫ И СВЯЗАННЫЕ РАБОТЫ

Давайте введем общее правило обновления для рассматриваемых методов как

$$w_{t+1} = w_t - \gamma_t B_t^{-1} m_t, \tag{3}$$

где γ_t — размер шага, $H_t = B_t^{-1}$ — специальная предобуславливающая матрица, и m_t обозначает либо g_t (градиент или его некоторая аппроксимация), либо первый момент градиента с параметром β_1. Для объяснения этого обновления мы можем представить, что направление спуска m_t шкалируется и вращается предобуславливающей матрицей H_t, и делается шаг с размером шага γ_t. Некоторые известные адаптивные методы первого порядка пользуются слегка упрощенной формой того же правила обновления:

$$w_{t+1} = w_t - \gamma_t m_t / v_t, \tag{4}$$

где m_t и v_t — первый и второй моменты, а m_t/v_t — покоординатное деление. Упомянутые типы шагов заключают в себе одну и ту же идею предобуславливания направления спуска и могут быть для простоты использованы взаимозаменяемо на протяжении всей статьи.

Таким же образом можно описать классические методы оптимизации. Например, для получения обновления SGD требуется обозначить предобуславливающую матрицу $B_t = I$, первый момент $m_t = g_t$ и размер шага γ_t как константу. Стоить отметить, что γ_t в SGD является особенно важным гиперпараметром, который требует специальной настройки в соответствии с заданными данными и функцией потерь, а методы с адаптивным размером шага, некоторые из которых используют предобуславливающую матрицу, основанную на локальной кривизне функции потерь, были представлены для устранения этой проблемы.

Классические методы с размером шага им. Б.Т. Поляка не используют такую информацию, но, тем не менее, стоит умопянуть о том, как получить классический детерминистический размер шага им. Б.Т. Поляка. Рассмотрим выпуклую функцию $f(w)$ и ограниченное сверху расстояние от w_{t+1} до оптимальной точки w^*:

$$\left\|w_{t+1} - w^*\right\|^2 \le Q(\gamma), \quad \text{где} \quad Q(\gamma) = \left\|w_t - w^*\right\|^2 - 2\gamma\left[f(w_t) - f^*\right] + \gamma_t^2 \left\|g_t\right\|_*^2.$$

Здесь g_t обозначает субградиент функции $f(w)$, а f^* — минимум функции. Минимизируя верхнюю границу $Q(\gamma)$, мы получаем размер шага им. Б.Т. Поляка и можем выразить его через правило обновления (3):

$$\gamma_t = \operatorname*{argmin}_{\gamma\in\mathbb{R}}\left[Q(\gamma)\right] = \frac{f(w_t) - f^*}{\left\|g_t\right\|_*^2}, \quad B_t = I \quad \text{и} \quad m_t = g_t. \tag{5}$$

Подробный разбор доказательства приведен в [26]. Заметим, что размер шага (5) может быть применен только в том случае, когда оптимальное значение f^* уже известно. Несмотря на то что иногда это значение

известно как $f^* = 0$ (например в задачах классификации), детерминистическая природа данного метода делает его непрактичным. Для решения этой проблемы был представлен стохастический градиентный спуск с размером шага им. Б. Т. Поляка (SPS, Stochastic Gradient Descent with Polyak Step-size) (см. [17]) вместе с более практичной версией $\mathrm{SPS}_{\max}$, который ограничивает γ_t постоянной γ_b:

$$\gamma_t^{\mathrm{SPS}} = \frac{f_i(w_t) - f_i^*}{\left\|\nabla f_i(w_t)\right\|_*^2} \quad \text{и} \quad \gamma_t^{\mathrm{SPSmax}} = \min\left\{\frac{f_i(w_t) - f_i^*}{\left\|\nabla f_i(w_t)\right\|_*^2}, \gamma_b\right\}. \tag{6}$$

Метод SPS все еще требует знания f_i^*, но при определенных режимах оптимизации стандартной нерегуляризированной функции потерь, таких как квадратичная задача для линейной регрессии и логистическая регрессия для классификации, оптимальное решение f_i^* равно 0. Если $f^* = 0$, то правило обновления SPS выражается как

$$\gamma_t = \frac{f_i(w_t)}{\| \nabla f_i(w_t) \|_*^2}, \quad H_t = I \quad \text{и} \quad m_t = \nabla f_i(w_t). \tag{7}$$

Также существует другой способ получения метода SPS. Если предположить, что выполнено условие интерполяции, то мы можем решить (1) путем выборки $i \in \{1, 2,..., n\}$ н.о.р.с.в. на каждой итерации t и решением нелинейного уравнения

$$w_{t+1} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \| w - w^t \|^2 \quad \text{т.ч.} \quad f_i(w) = 0. \tag{8}$$

Хотя приведенная выше проекция может иметь аналитическое решение для некоторых простых функций потерь, для большинства нелинейных моделей, таких как глубокие нейронные сети, не существует решения в замкнутой форме. Поэтому вместо точного решения мы можем линеаризовать $f_i(w)$ вокруг текущей итерации w^t, чтобы получить

$$w_{t+1} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left\|w - w^t\right\|^2 \quad \text{т.ч} \quad f_i(w^t) + \left\langle \nabla f_i(w^t), w - w^t \right\rangle = 0.$$

Правило обновления (7) и есть аналитическое решение этой задачи.

Вне режима интерполяции решение для (8) может не существовать. Поэтому вместо того, чтобы пытаться обнулить все функции потерь, мы можем попытаться приблизить их к нулю, минимизировав дополнительную переменную остатка (slack) следующим образом:

$$\underset{w \in \mathbb{R}^d, s \geq 0}{\operatorname{argmin}} \quad s$$

$$\text{т.ч.} \quad f_i(w) \leq s \quad \text{для} \quad i = 1,\ 2,\ldots,\ n;$$

$$\underset{w \in \mathbb{R}^d, s \geq 0}{\operatorname{argmin}} \quad s^2$$

$$\text{т.ч.} \quad f_i(w) \leq s \quad \text{для} \quad i = 1,\ 2,\ldots,\ n;$$

которые называются *L1*- и *L2*-остаточными минимизациями (см. [19]) соответственно. Отметим, что цель этого метода состоит в том, чтобы приблизить s к нулю, что позволяет решать задачи, в которых предположение интерполяции не выполняется.

3. РЕЗУЛЬТАТЫ

В статье мы объединяем предобусловливание и варианты остаточно-регуляризованных методов SPS. Затем мы демонстрируем, что эти новые предобусловленные методы хорошо работают на плохо масштабированных и плохо обусловленных данных.

• **Усовершенствованный SPS.** Мы расширили методы SPS и представили три новых алгоритма: PSPS, PSPSL1 и PSPSL2, которые используют метод Хатчинсона, Adam и AdaGrad для предобусловливания градиентного шага с использованием размера шага им. Б. Т. Поляка для взвешенной евклидовой нормы. Правила обновлений наших методов в явном виде описаны ниже.

• **Имплементация в PyTorch.** Мы разработали практические варианты наших методов в качестве оптимизаторов PyTorch и опубликовали программный код в нашем репозитории GitHub[1].

• **Эмпирические Результаты.** Мы привели несколько экспериментов с двумя разными задачами, чтобы сравнить наши результаты с SGD, Adam, AdaGrad и с вариантами SPS, в которых не применяются какие-либо методы предобусловливания. Мы показали, что предложенные нами алгоритмы демонстрируют заметные улучшения на плохо обусловленных задачах.

4. ПРЕДОБУСЛОВЛИВАНИЕ

Данные могут быть плохо масштабированы и/или плохо обусловлены, тогда предобусловливание градиента — это один из способов улучшить сходимость алгоритмов. Методы, использующие предобусловливание, имеют следующее общее правило обновления:

$$w_{t+1} = w_t - \gamma_t B_t^{-1} \nabla f_i(w_t),$$

где $B_t \in \mathbb{R}^{d\times d}$ — обратимая и положительно-определенная матрица. Метод Ньютона — один из самых наглядных примеров метода, использующего предобусловливание. В этом случае $B_t = \nabla^2 f(w_t)$ и $\gamma_t = 1$. Среди более современных и практичных методов с предобуславливанием отметим AdaHessian [27], Adagrad [6] и OASIS [28]. Эти методы включают кривизну функции потерь посредством адаптивных оценок Гессиана.

4.1. Метод Хатчинсона

Метод Хатчинсона (см. [25]) используется для оценки диагонали матрицы Гессиана. Для вычисления этой оценки метод Хатчинсона использует лишь несколько произведений Гессиана на вектор, которые, в свою очередь, можно эффективно вычислить с помощью быстрого автоматического дифференцирования (см. [29]). Произведение матрицы Гессиана $\nabla^2 f(w)$ и фиксированного вектора z можно вычислить через производную градиента по направлению. Чтобы понять, как этот метод используется для предобусловливания, сначала мы покажем, что затраты на вычисление произведения Гессиана на вектор близки к двух вычислениям градиентов, т.е.

$$\nabla^2 f(w)z = \nabla(z^T \nabla f(w)). \tag{9}$$

Затем мы можем вычислить диагональ Гессиана, используя метод Хатчинсона:

$$\operatorname{diag}(\nabla^2 f(w)) = \mathbb{E}\Big[z \odot (\nabla^2 f(w)z)\Big],$$

где z — случайный вектор с распределением Радемахера[2] или нормальным распределением, а $\nabla^2 f(w)z$ вычисляется с помощью произведения Гессиана на вектор, заданного в (9).

Можно доказать, что математическое ожидание $z \odot (\nabla^2 f(w)z)$ является диагональю Гессиана (см. [30]). Используя это тождество, мы оцениваем диагональ Гессиана по заданному D_0, генерируя случайный вектор z на каждой итерации и обновляя нашу оценку с использованием средневзвешенного значения следующим образом:

$$D_t = \beta D_{t-1} + (1-\beta)\operatorname{diag}(z \odot \nabla^2 f(w)z),$$

где $\beta \in (0,1)$ — параметр момента и

$$D_0 = \frac{1}{m}\sum_{i=1}^{m} \operatorname{diag}(z_i \odot \nabla^2 f(w_0)z_i).$$

Наконец, чтобы гарантировать, что D_t остается положительно-определенным, несмотря на возможную невыпуклость функций потерь, мы используем усечение и сохраняем только абсолютные значения элементов следующим образом: $(\hat{D}_t)_{j,j} = \max\{\alpha, |D_t|_{j,j}\}$.

Algorithm 1. Аппроксимация диагонали Гессиана с использованием метода Хатчинсона

1: **Ввод:** $\beta \in (0,1)$, $\alpha > 0$
2: **Инициализация:** $D_0 = \frac{1}{m}\sum_{i=1}^{m} \operatorname{diag}(z_i \odot \nabla^2 f(w_0)z_i)$
3: **for** $t = 1, 2, \ldots, T$ **do**

[1] https://github.com/fxrshed/ScaledSPS.
[2] $z_i \in \{-1, +1\}$ с равной вероятностью.

4: Генерируем случайный вектор z из Радемахера/нормального распределения

5: $D_t = \beta D_{t-1} + (1-\beta)\,\mathrm{diag}(z \odot \nabla^2 f(w_0)z)$

6: $(\hat{D}_t)_{j,j} = \max\{\alpha, |\, D_t\,|_{j,j}\}$

7: **Вывод:** $\hat{D}_T$

4.2. Метод AdaGrad

AdaGrad — это метод стохастической оптимизации, который аппроксимирует Гессиан функции, чтобы адаптировать размер шага в зависимости от информации о кривизне. Ключевая идея заключается в использовании информации о кумулятивном квадрате градиента для шкалирования размера шага. В форме (4) правило обновления для AdaGrad может быть задано следующим образом:

$$m_t = \mathbf{g_t} \quad \text{и} \quad v_t = \sqrt{\sum_{i=1}^{t} \mathbf{g_i} \odot \mathbf{g_i}}.$$

Накопление всех предыдущих градиентов в предобуславливателе v_t приводит к уменьшению размера шага γ_t, что повышает производительность при разреженных данных (нечастых признаках), при этом ухудшается в случае плотных данных.

4.3. Метод Adam

Представленный в [7] Adam разработан для преодоления недостатков других популярных алгоритмов оптимизации, таких как AdaGrad [6] и RMSProp [31], путем включения как адаптивного размера шага, так и обновлений на основе метода “тяжелого шарика” (momentum). Правило обновления Adam предполагает вычисление скользящего среднего как для первого, так и для второго моментов градиентов. Первый момент — это среднее значение градиентов, а второй момент — нецентрированная дисперсия градиентов. Правило обновления для Adam может быть выражено в терминах (4) следующим образом:

$$m_t = \frac{(1-\beta_1)\sum_{i=1}^{t}\beta_1^{t-i}\mathbf{g_i}}{1-\beta_1^t}, \quad v_t = \sqrt{\frac{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\mathbf{g_i} \odot \mathbf{g_i}}{1-\beta_2^t}},$$

где $0 < \beta_1, \beta_2 < 1$ — два гиперпараметра, называемых коэффициентами первого и второго моментов. Смещенные оценки корректируются путем деления их на члены коррекции смещения, которые являются степенями скоростей затухания β_1 и β_2 соответственно.

5. ПРЕДОБУСЛОВЛЕННЫЙ СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК С РАЗМЕРОМ ШАГА им. Б. Т. ПОЛЯКА

В этом разделе мы предлагаем новые методы, основанные на ранее описанных, таких как SPS. Прежде всего, чтобы описать их, мы рассмотрим задачу проекции на множество ограничений

$$w_{t+1} = \operatorname*{argmin}_{w \in \mathbb{R}^d} \|w - w_t\|^2 \quad \text{т.ч.} \quad f_i(w) = 0. \tag{10}$$

Обратите внимание, что ограничение $f_i(w) = 0$ определено как условие интерполяции.

Определение 1. Мы предполагаем, что условие интерполяции выполняется для набора функций $\{f_i(w)\}_{i=1}^n$ по заданному набору данных $\{(x_i, y_i)\}_{i=1}^n$ с неотрицательными функциями потерь $f_i(w) \geq 0$, когда

$$\exists w^* \in \mathbb{R}^d \quad \text{т.ч.} \quad f_i(w^*) = 0 \quad \forall i \in \{1, 2, \ldots, n\}.$$

Одним из представленных методов, используемых в настоящей работе, является использование предобуславливания для улучшения скорости сходимости в случае плохо обусловленных данных. Чтобы получить это, мы изменяем норму в проекции (10) на взвешенную норму, основанную на предобуславливателе $B_t \succ 0$. Другой важной частью является линейная аппроксимация условия интерполяции $f_i(w) = 0$. Согласно разложению Тейлора функции $f_i(w)$, линейное приближение (первого порядка) задается через $f_i(w) \approx f_i(w_t) + \langle \nabla f_i(w_t), w - w_t \rangle$. Мы используем это приближение, чтобы ослабить условие интерполяции, которое не допускает решения в явном виде для большинства нелинейных моделей. Другой способ получения аналитического решения — ввести дополнительную переменную остатка (описано позже).

Предобусловленный SPS. Мы рассматриваем дифференцируемую выпуклую функцию f_i и линеаризацию условия интерполяции. Чтобы вывести предобусловленное правило обновления, мы используем взвешенную норму в проекции, полученный метод мы называем PSPS (Preconditioned Stochastic Gradient Descent with Polyak Step-size). В настоящей статье мы рассмотрим три варианта предобуславливания, а именно, метод Хатчинсона и предобуславливание оптимизаторов AdaGrad и Adam.

Лемма 1 (PSPS). *Пусть $B_t \succ 0$ для всех $t \geq 0$, тогда итеративный явный шаг для задачи*

$$w_{t+1} = \arg\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w - w_t\|^2_{B_t}, \quad m.ч. \quad f_i(w_t) + \langle \nabla f_i(w_t), w - w_t\rangle = 0$$

выражается как

$$w_{t+1} = w_t - \frac{f_i(w_t)}{\|\nabla f_i(w_t)\|^2_{B_t^{-1}}} B_t^{-1}\nabla f_i(w_t).$$

Отметим, что данный шаг может быть переформулирован в виде шага (3), где

$$\gamma_t = \frac{f_i(w_t)}{\|\nabla f_i(w_t)\|^2_{B_t^{-1}}} \quad \text{и} \quad m_t = \nabla f_i(w_t).$$

Аналогичным образом мы можем применить предобуславливание для методов с остатком и получить следующие два метода: PSPSL1 и PSPSL2.

Лемма 2 (PSPSL1). *Пусть $B_t \succ 0$ для любых $t \geq 0$ и $\mu, \lambda > 0$, тогда явный вид шага для задачи*

$$w_{t+1}, s_{t+1} = \arg\min_{w\in\mathbb{R}^d, s\geq 0} \tfrac{1}{2}\| w - w_t \|^2_{B_t} + \mu(s - s_t)^2 + \lambda s,$$
$$m.ч. \quad f_i(w_t) + \langle \nabla f_i(w_t), w - w_t\rangle \leq s$$

выражается как

$$\gamma_t^{L1} = \frac{(f_i(w_t) - s_t + \lambda/2\mu)_+}{1/2\mu + \|\nabla f_i(w_t)\|^2_{B_t^{-1}}}, \quad \gamma_t = \min\left\{\gamma_t^{L1}, \frac{f_i(w_t)}{\|\nabla f_i(w_t)\|^2_{B_t^{-1}}}\right\},$$

$$w_{t+1} = w_t - \gamma_t B_t^{\ 1}\nabla f_i(w_t), \quad s_{t+1} = (s_t - \tfrac{1}{2\mu}(\lambda + \gamma_t^{L1}))_+.$$

Лемма 3 (PSPSL2). *Пусть $B_t \succ 0$ для любых $t \geq 0$ и $\mu, \lambda > 0$, тогда явный вид решения задачи*

$$w_{t+1}, s_{t+1} = \arg\min_{w\in\mathbb{R}^d, s\in\mathbb{R}} \| w - w_t \|^2_{B_t} + \mu(s - s_t)^2 + \lambda s^2,$$
$$m.ч. \quad f_i(w_t) + \langle \nabla f_i(w_t), w - w_t\rangle \leq s \tag{11}$$

выражается как

$$w_{t+1} = w_t - \frac{(f_i(w_t) - \mu\hat{\lambda}s_t)_+}{\hat{\lambda} + \|\nabla f_i(w_t)\|^2_{B_t^{-1}}} B_t^{-1}\nabla f_i(w_t),$$

$$s_{t+1} = \hat{\lambda}\left(\mu s_t + \frac{(f_i(w_t) - \mu\hat{\lambda}s_t)_+}{\hat{\lambda} + \|\nabla f_i(w_t)\|^2_{B_t^{-1}}}\right),$$

где $\hat{\lambda} = 1/(\mu + \lambda)$. Здесь остаточная параметр λ заставляет s быть ближе к 0, пока μ не дает s_{t+1} быть далеко от s_t.

6. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В этом разделе мы представляем эксперименты, проведенные с использованием предложенных нами методов и некоторых из наиболее популярных оптимизаторов: SGD, Adam и AdaGrad. Выбор этих методов оправдан тем фактом, что все они, за исключением SGD, используют адаптивный размер шага. В наших экспериментах каждый из этих методов представлен с разными размерами шага, чтобы показать разницу в сходимости.

Мы использовали датасеты из LIBSVM[3], а именно, mushrooms и colon-cancer, для иллюстрации эффективности предложенных методов, минимизирующих функцию потерь логистической регрессии и нелинейных наименьших

[3] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

квадратов в задачах бинарной классификации. Кроме того, каждый эксперимент дополнительно проводится на плохо обусловленной версии тех же наборов данных, где столбцы умножаются на вектор $e = \{\exp(x_i)\}_{i=1}^{d}$, где x_i генерируется из равномерного распределения с интервалом $[-k, k]$. На всех приведенных далее иллюстрациях термин k относится к этому коэффициенту шкалирования, где $k = 0$ — исходные данные.

Во время обучения предложенными методами мы применяли параметры остатка $\lambda = 0.01$ и $\mu = 0.1$. Для метода Хатчинсона мы применили $\alpha = 10^{-4}$ и $\beta = 0{,}999$. Гиперпараметры (за исключением размера шага) для других методов (SGD, Adam и т.д.) были сохранены в качестве значений по умолчанию. Все эксперименты проводились с пятью различными ключами генераторов случайности (seed), используя PyTorch 1.11.0.

Оптимизируемые функции. Пусть $\{(x_i, y_i)\}_{i=1}^{n}$ — это данные из выбранного датасета. Логистическая регрессия определена следующим образом:

$$f_{\text{LogReg}}(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^{\mathrm{T}} w)),$$

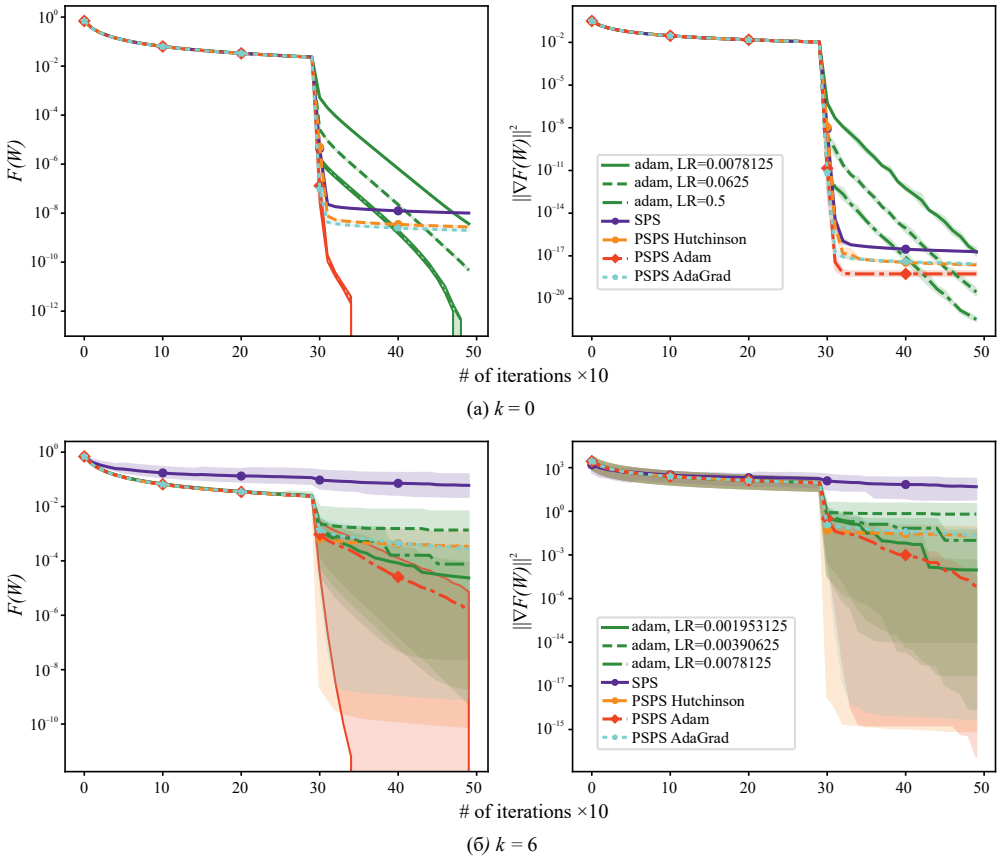
где $x_i \in \mathbb{R}^d$ и $y_i \in \{-1, +1\}$. Нелинейные наименьшие квадраты заданы как

$$f_{\text{NLLSQ}}(w) = \frac{1}{n}\sum_{i=1}^{n} (y_i - 1/(1 + \exp(-x_i^{\mathrm{T}} w)))^2,$$

где $y_i \in \{0, 1\}$.

На фиг. 3 мы сравниваем скорости сходимости SPS с предобуславливателем Adam и без него. Наблюдаем, что в случае плохо обусловленных данных нам необходимо точно настроить размер шага оптимизатора Adam, чтобы избежать расхождений, поскольку выбор одинаковых размеров шага в обоих версиях данных привело к расхождению с $k = 6$. Кроме того, мы можем наблюдать, как различные методы предобуславливания превосходят SPS без какого-либо предобуславливания как для исходных данных, так и для плохо обусловленных. Отсутствие необходимости ручной настройки размера шага является одним из преимуществ предобусловленных методов SPS. Аналогичные результаты можно наблюдать на фиг. 6 и 9 для датасета colon-cancer. На фиг. 3б мы видим, что шкалирование данных приводит к тому, что размер шага Adam приходится уменьшать по мере увеличения коэффициента шкалирования k, чтобы метод не расходился.

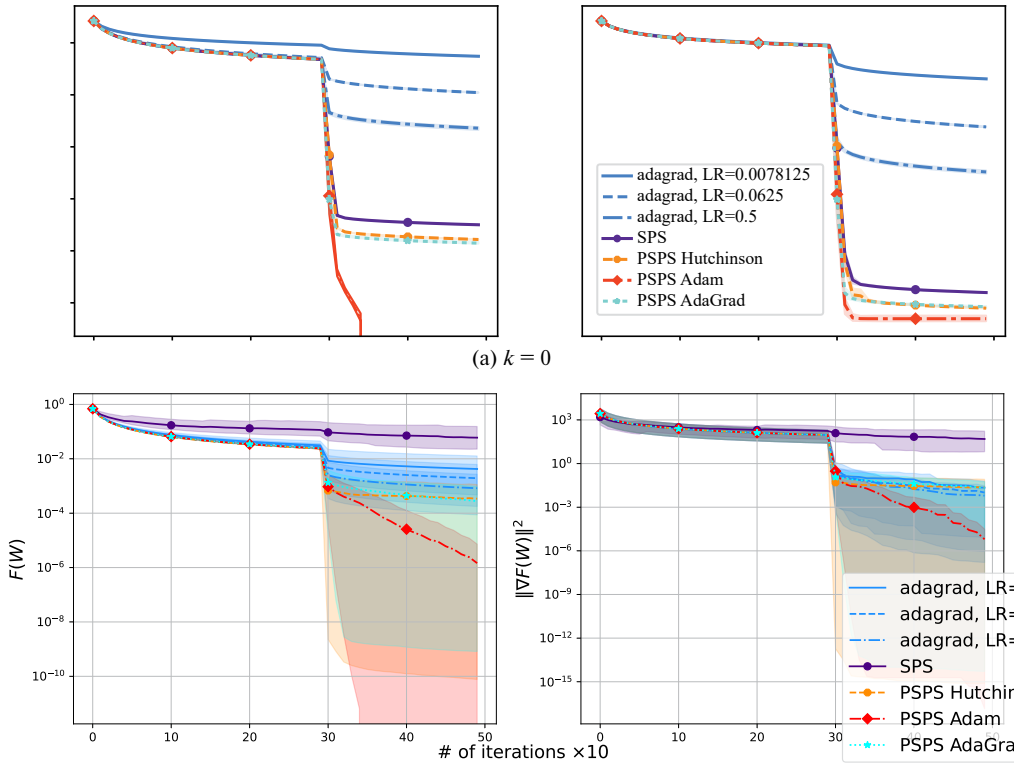
Мы также сравниваем наши методы с оригинальными SPS, SPSL1, SPSL2, SGD и Adam (фиг. 4 и 5).



(а) $k = 0$

(б) $k = 6$

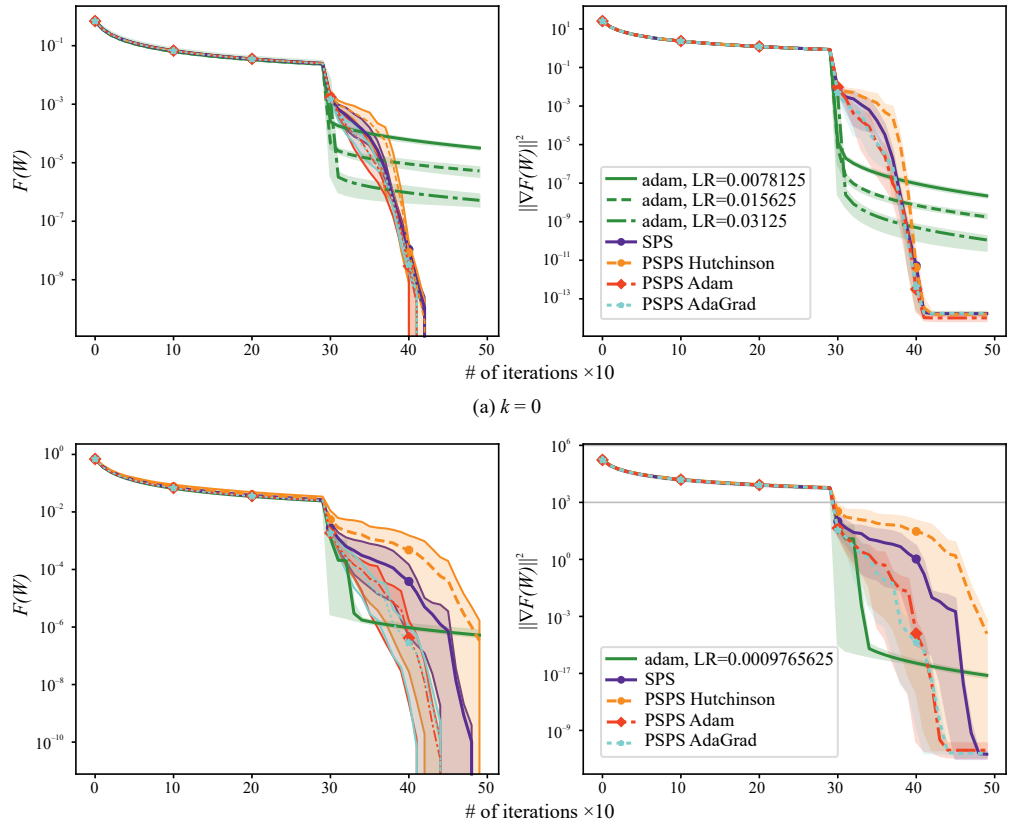
Фиг. 1. Метод Adam vs PSPS с разным предобуславливанием для логистической регрессии на датасете mushrooms.



(а) $k = 0$

(б) $k = 6$

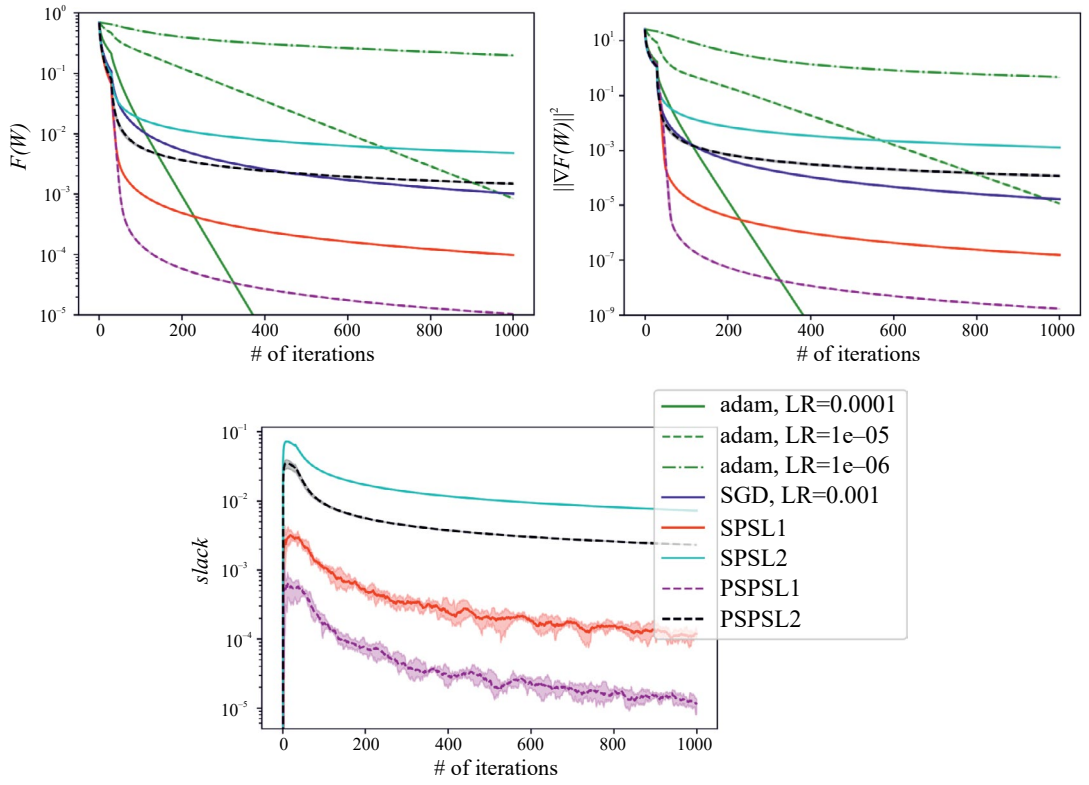
Фиг. 2. Методы AdaGrad vs PSPS с разным предобуславливанием для логистической регрессии на датасет mushrooms.



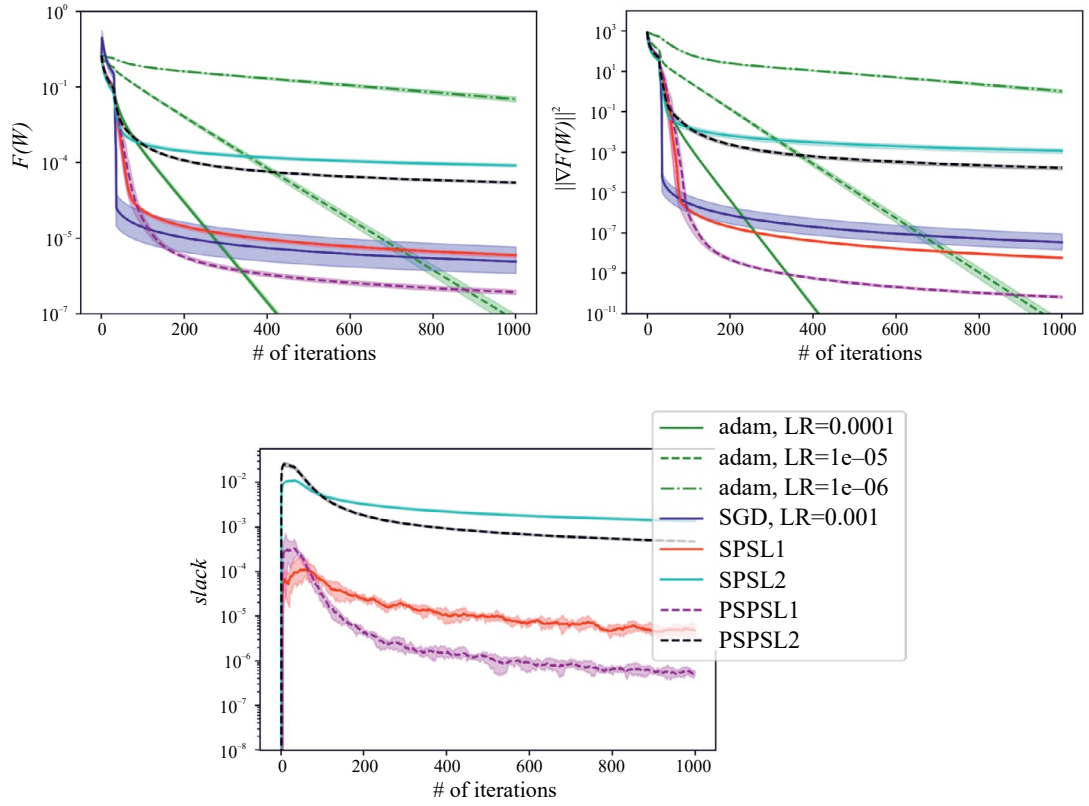
(а) $k = 0$

(б) $k = 6$

Фиг. 3. Методы Adam vs PSPS с разным предобуславливанием для логистической регрессии на датасете colon-cancer.

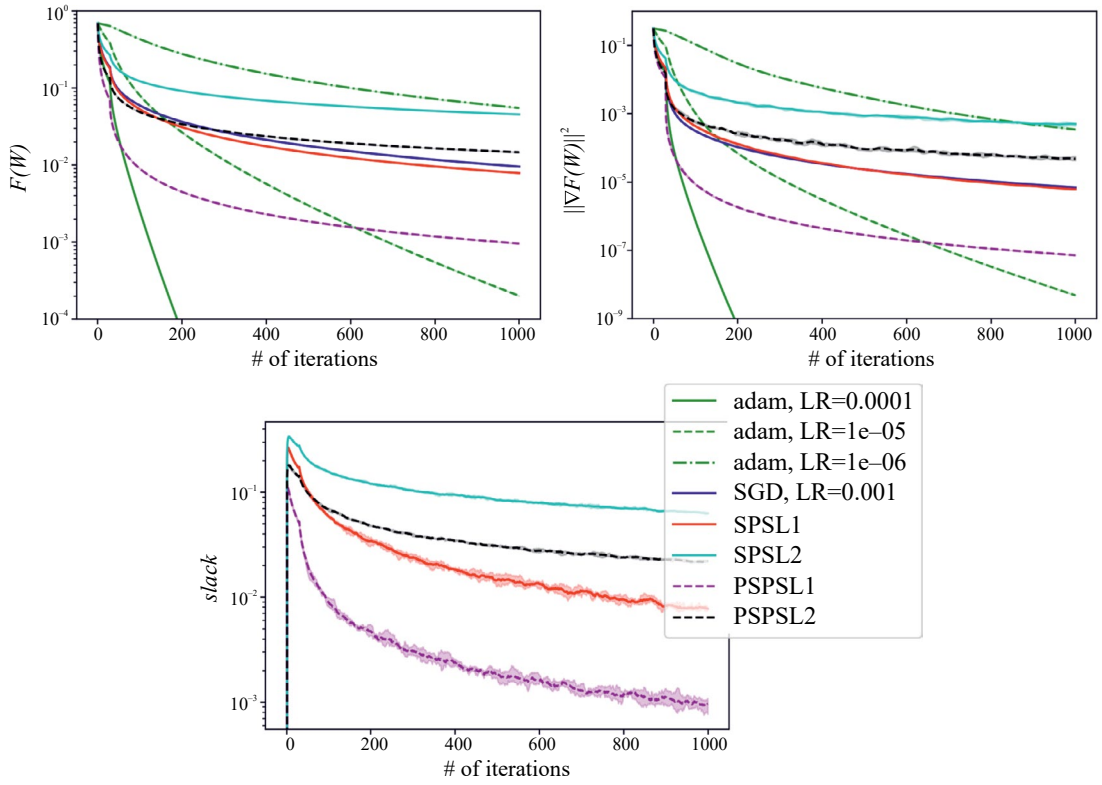


(а) $k = 0$

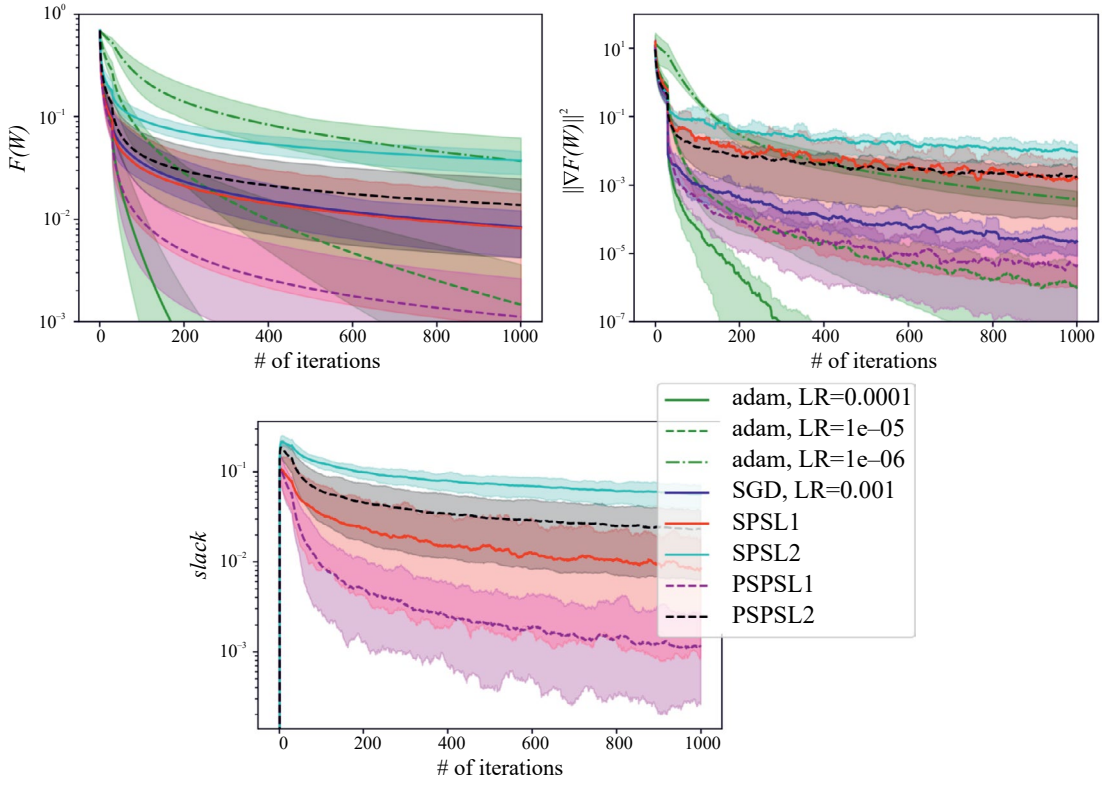


(б) $k = 3$

Фиг. 4. Сравнение эффективности PSPSL1 и PSPSL2 с SPS, SGD и Adam для логистической регрессии на оригинальных и плохо обусловленных версиях датасета colon-cancer.



(а) $k = 0$



(б) $k = 3$

Фиг. 5. Сравнение эффективности PSPSL1 и PSPSL2 с SPS, SGD и Adam для логистической регрессии на оригинальных и плохо обусловленных версиях датасета mushrooms.

6. ЗАКЛЮЧЕНИЕ

В статье мы изучили влияние предобуславливания на семейство методов SPS (стохастический градиентный спуск с размером шага им. Б. Т. Поляка). Мы предложили новые методы PSPS, PSPSL1, PSPSL2 в (11)–(13). Эксперименты проводились как в выпуклых, так и в невыпуклых случаях с двумя разными датасетами. В настоящей статье отсутствует теоретический анализ предлагаемых нами методов, который может быть проведен в качестве последующей исследовательской работы. Кроме того, интересно провести эксперименты с более сложными моделями, такими как глубокие нейронные сети.

СПИСОК ЛИТЕРАТУРЫ

1. Bekas C., Kokiopoulou E., Saad Y. An estimator for the diagonal of a matrix // Appl. Numer. Math. 2007. V. 57. № 11. P. 1214—1229.
2. Berrada L., Zisserman A., Kumar M. P. Training neural networks for and by interpolation. In Hal Daumé III and Aarti Singh, eds. // Proceed. 37th Inter. Conf. Mach. Learn. 2020. V. 119. P. 799—809.
3. Boyd S., Xiao L., Mutapcic A. Subgradient methods. lecture notes of EE392o, Stanford Univer., Autumn Quarter. 2023. V. 2004. P. 2004—2005.
4. Christianson B. Automatic Hessians by reverse accumulation // IMA J. Numer. Analys. 1992. V. 12. № 2. P. 135—150.
5. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization // J. Mach. Learn. Res. 2011. V. 12. № 61. P. 2121—2159.
6. Garrigos G., Gower R. M., Schaipp F. Function value learning: Adaptive learning rates based on the polyak stepsize and function splitting in erm // arXiv preprint arXiv:2307.14528, 2023.
7. Gower R.M., Blondel M., Gazagnadou N., Pedregosa F. Cutting some slack for sgd with adaptive polyak stepsizes // arXiv preprint arXiv:2202.12328, 2022.
8. Hutchinson M.F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines // Comm. in Statistics-Simulation and Computat. 1989. V. 18. № 3. P. 1059—1076.
9. Jahani M., Rusakov S., Shi Zh., Richtárik P., Mahoney M. W., Takáč M. Doubly adaptive scaled algorithm for machine learning using second-order information // In 10th Inter. Conf. Learn. Representat. (ICLR2022), 2022.
10. Jiang X., Stich S. U. Adaptive sgd with polyak stepsize and line-search: Robust convergence and variance reduction // arXiv preprint arXiv:2308.06058, 2023.
11. Kingma D., Ba J. Adam: A method for stochastic optimization // Inter. Conf. Learn. Representat. (ICLR), San Diego, CA, USA, 2015.
12. Lan G. An optimal method for stochastic composite optimization // Math. Program. 2012. V. 133. P. 365—397.
13. Li Sh., Swartworth W. J., Takáč M., Needell D., Gower R. M. SP2: A second order stochastic polyak method // 11th Inter. Conf. on Learn. Representat., 2023.
14. Li X., Orabona F. On the convergence of stochastic gradient descent with adaptive stepsizes. In Kamalika Chaudhuri and Masashi Sugiyama, eds. // Proceed. 22nd Inter. Conf. Artific. Intelligence and Statistic. 2019. V. 89. P. 983—992.
15. Loizou N., Vaswani Sh., Laradji I. H., Lacoste-Julien S. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In Arindam Banerjee and Kenji Fukumizu, eds. // Proceed. 24th Inter. Conf. Artific. Intelligence and Statistic. 2021. V. 130. P. 1306—1314.
16. Loshchilov I., Hutter F. Decoupled weight decay regularization // Inter. Conf. Learn. Representat., 2019.
17. Nemirovski A., Juditsky A., Lan G., Shapiro A. Robust stochastic approximation approach to stochastic programming // SIAM J. Optimizat. 2009. V. 19. № 4. P. 1574—1609.
18. Orvieto A., Lacoste-Julien S., Loizou N. Dynamics of sgd with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds. // Adv. Neural Informat. Proces. System. 2022. V. 35. P. 26943—26954.
19. Polyak B.T., Juditsky A. B. Acceleration of stochastic approximation by averaging.
20. SIAM J. Control and Optimizat. 1992. V. 30. № 4. P. 838—855.
21. Polyak B. T. Minimization of unsmooth functionals // USSR Comput. Math. and Math. Phys. 1969. V. 9. P. 14—29.
22. Polyak B. T. Introduction to optimization. Optimization Software, Inc., Publ. Division, 1987.
23. Polyak B.T. A new method of stochastic approximation type // Avtomatika i Telemekhanika. 1990. V. 51. P. 98—107.
24. Reddi S.J., Kale S., Kumar S. On the convergence of adam and beyond // Inter. Conf. Learn. Representat., 2018.
25. Robbins H., Monro S. A stochastic approximation method // Ann. Math. Statistic. 1951. V. 22. P. 400—407.

26. Sadiev A., Beznosikov A., Almansoori A. J., Kamzolov D., Tappenden R., Takáč M. Stochastic gradient methods with preconditioned updates // arXiv preprint arXiv:2206.00285, 2022.
27. Schaipp F., Gower R. M., Ulbrich M. A stochastic proximal polyak step size // arXiv preprint arXiv:2301.04935, 2023.
28. Schaipp F., Ohana R., Eickenberg M., Defazio A., Gower R. M. Momo: Momentum models for adaptive learning rates // arXiv preprint arXiv:2305.07583, 2023.
29. Shi Zh., Sadiev A., Loizou N., Richtárik P., Takáč M. AI-SARAH: Adaptive and implicit stochastic recursive gradient methods // Transact. Mach. Learn. Res., 2023.
30. Tieleman T., Hinton G., et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude // COURSERA: Neural Networks for Machine Learn. 2012. V. 4. № 2. P. 26—31.
31. Ward R., Wu X., Bottou L. Adagrad stepsizes: Sharp convergence over nonconvex landscapes // J. Mach. Learn. Res. 2020. V. 21. № 1. P. 9047—9076.
32. Yao Zh., Gholami A., Shen Sh., Mustafa M., Keutzer K., Mahoney M. Adahessian: An adaptive second order optimizer for machine learning // Proceed. AAAI Conf. Artific. Intelligence. 2021. V. 35. P. 10665—10673.

STOCHASTIC GRADIENT DESCENT WITH PRECONDITIONED POLYAK STEP-SIZE

F. Abdukhakimov[a,*], Ch. Xiang[a,], D. Kamzolov[a,**], M. Takáč[a,****]**

[a] *Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE*
[*] *e-mail: farshed888@gmail.com*
[**] *e-mail: chulu.xiang@mbzuai.ac.ae*
[***] *e-mail: kamzolov.opt@gmail.com*
[****] *e-mail: takac.mt@gmail.com*

Received 02 November, 2023
Revised 16 December, 2023
Accepted 20 December, 2023

Abstract. Stochastic Gradient Descent (SGD) is one of the many iterative optimization methods that are widely used in solving machine learning problems. These methods display valuable properties and attract researchers and industrial machine learning engineers with their simplicity. However, one of the weaknesses of this type of methods is the necessity to tune learning rate (step-size) for every loss function and dataset combination to solve an optimization problem and get an efficient performance in a given time budget. Stochastic Gradient Descent with Polyak Step-size (SPS) is a method that offers an update rule that alleviates the need of fine-tuning the learning rate of an optimizer. In this paper, we propose an extension of SPS that employs preconditioning techniques, such as Hutchinson's method, Adam, and AdaGrad, to improve its performance on badly scaled and/or ill-conditioned datasets.

Keywords: machine learning, optimization, adaptive step-size, polyak step-size, preconditioning.