Адаптивный стохастический градиентный спуск

Евгений Лагутин Daniil Merkulov
lagutin.em@phystech.edu daniil.merkulov@skoltech.ru

Project Proposal

Проект подразумевает реализацию метода адаптивного стохастического спуска, который можно рассматривать как модификацию универсального градиентного спуска (в смысле схожести подбора константы Липшица градиента ([1])). Предполагается, что метод имеет хорошее качество на практике, однако подтвердить это теоретически пока не удалось.

1 Идея

Конструкция минибатчинга позволяет переносить оптимальные методы на задачи стохастической оптимизации с сохранением свойства оптимальности и получать оптимальные методы для задач стохастической оптимизации из неоптимальных методов для детерминированных задач. Также в большинстве приложений нельзя сделать предположений о конкретных свойствах функций, таких как значения константы Липшица градиента и дисперсии градиента. Поэтому используются адаптивные стохастические методы. В данном проекте реализуется один из таких методов. Хотя оптимальность данного метода пока не доказана, хочется проверить, насколько хорошо он работает на практике. Задача состоит в том, чтобы проверить качество метода на различных классах функций и сравнить с качеством работы существующих методов (выявив, таким образом, для каких классов метод может оказаться оптимальным). Основной упор предлагается сделать в исследовании качества работы метода на функциях вида суммы большого числа функционалов, часто возникающих в задач машинного обучения и глубокого обучения. Для таких функций конструкция стохастического усреднения легко переносится на конструкцию рандомизации суммы ([2]), позволяя очевидным образом применить стратегию минибатчинга, резко улучшающую качество метода за счёт распараллеливания вычислений.

1.1 Problem

Рассмотрим следующую задачу выпуклой оптимизации:

Вместо градидиента $\nabla f(x)$ оракул выдает его несмещённую оценку $\nabla_x f(x, \xi)$ с конечной дисперсией.

$$\mathbb{E}_\xi[\nabla_x f(x, \xi)] \equiv \nabla f(x),\ \mathbb{E}[\|\nabla_x f(x, \xi) - \nabla f(x)\|_2^2] < D \tag{1}$$

Конструкция минибатчинга в общем представлении:

$$\overset{r}{\nabla}_x f(x, \{\xi^l\}_{l=1}^r) = \frac{1}{r}\sum_{l=1}^{r} \nabla_x f(x, \xi^l) \tag{2}$$

В [1] показано, что для выпуклых функций, обладающих липшецевым градиентом, т.е. таких, что $|\nabla f(x) - \nabla f(y)| \leq L\|x - y\|_2$, справедлива следующая оценка на количество обращений к оракулу при не малых D будет

$$N(\varepsilon) = O\left(\frac{DR^2}{\varepsilon^2}\right), \tag{3}$$

причём эта оценка остаётся верной и для ускоренных методов и не является улучшаемой [3].

Для сильно выпуклого случая данную оценку можно улучшить при помощи конструкции рестартов. В результате получим неулучшаемую оценку для сильно выпуклого случая:

$$N(\varepsilon) = O\left(\min\left\{\frac{DR^2}{\varepsilon^2}, \frac{D}{\mu\varepsilon}\right\}\right), \tag{4}$$

В случае, когда неизвестны значения L и D (считая при этом, что сделанные выше предположения выполнены), используется адаптивный метод подбора константы L.

В данной работе предлагается исследовать следующий адаптивный алгоритм:

Algorithm 1 Adaptive Stochastic Gradient Method (Spokoiny's practical variant)

Input: lower estimate for the variance of the gradient $D_0 \leq D$,
accuracy $0 < \varepsilon < \frac{D_0}{L}$, starting point $x_0 \in Q$, initial guess $L_{-1} > 0$

for $k = 0, 1, ...$ **do**
  Set $i_k = 0$. Set $r^k = \lceil \frac{2D_0}{L_{k-1}\varepsilon} \rceil$, generate i.i.d. $\xi_K^i,\ i = 1, \dots, r^k$
  repeat
    Set $L_k = 2^{i_k - 1} L_{k-1}$
    Calculate $\tilde{g}(x_k) = \frac{1}{r^k} \sum_{i=1}^{r^k} \nabla f(x_k, \xi_k^i)$.
    Calculate $w_k = x_k - \frac{1}{2L_k} \tilde{g}(x_k)$.
    Calculate $\tilde{f}(x_k) = \frac{1}{r_k} \sum_{i=1}^{r^k} f(x_k, \xi_k^i)$ and
    $\tilde{f}(w_k) = \frac{1}{r^k} \sum_{i=1}^{r^k} f(w_k, \xi_k^i)$.
    Set $i_k = i_k + 1$.
  until
    $\tilde{f}(w_k) \leq \tilde{f}(x_k) + \langle \tilde{g}(x_k), w_k - x_k \rangle + \frac{2L_k}{2} \|w_k - x_k\|_2^2 + \frac{\epsilon}{10}$.
  Set $x_{k+1} = w_k,\ k = k + 1$.
end for

Заметим, что практический вариант отличается от теоретического тем, что один и тот же набор случайных величин используется для подсчёта градиента для шага и для подсчёта значений функции для проверки удовлетворения L_k условию перехода к следующему шагу.

К сожалению, пока нет предположений о конкретных классах задач, где метод имеет лучшее качество, поэтому предлагается провести численное исследование. В частности, проверить качество метода на следующем виде задач.

Во многих задачах функционал имеет вид:

$$\frac{1}{m} \sum_{i=1}^{m} f(x_i) \to \min_{x \in Q} \tag{5}$$

Если m - большое число, равновероятно генерируется набор r слагаемых $\{\xi^l\}_{l=1}^r$, вычисляется следующая оценка градиента:

$$\nabla f(x, \{\xi^l\}_{l=1}^r) = \frac{1}{r} \sum_{l=1}^{r} \nabla f_{\xi^l}(x). \tag{6}$$

Такой подход называется рандомизацией суммы.

С другой стороны, эту конструкцию можно рассматривать как минибатчинг.

$$f(x) = \mathbb{E}_\xi[f(x, \xi)] \to \min_{x \in Q}, \quad f(x, \xi) = f_\xi(x), \quad \nabla_x f(x, \xi) = \nabla f_\xi(x), \tag{7}$$

$$P(\xi = l) = \frac{1}{m},\ l = 1, \dots, m.$$

2 Outcomes

Как уже было отмечено выше, результатом проекта будет реализация предложенного метода, а также набор численных экспериментов. Качество метода будет проверено на различных классах функций (невыпуклых, выпуклых, сильно выпуклых). Так для сильно выпуклого случая можно реализовать конструкцию рестартов, позволяющую переносить оптимальные оценки с выпуклого случая. Особое внимание будет уделено проверке метода на задачах машинного обучения и глубокого обучения (логистическая регрессия, двухслойная нейронная сеть на классических датасетах `MNIST`, `CIFAR10` и т.п.). Для таких функций градиенты будем вычислять параллельно. Также итогом будет сравнение с другими методами адаптивного стохастического градиентного спуска. Среди таких методов - различные варианты метода AdaGrad, Adam.

В работах, описанных ниже, помимо предложенных методов, оценок, теорем и прочего, есть описания численных экспериментов по проверке качества методов. Эта же техника будет использована мною для проверки качества предложенного метода. Будет выбран набор классических мировых датасетов (`MNIST`, `CIFAR10`, `IMDB`) и различные логистические модели: логистическая регрессия, полносвязная нейросеть, конволюционная нейросеть. Далее наша модель будет сравниваться с другими адаптивными стохастическими градиентными методами (AdaGrad, Adam, их вариации). Сравнение может происходить по следующим критериям: скользящее среднее функции потерь, скользящая точность, точность после 1 эпохи, точность после нескольких эпох.

В качестве примера можно привести график из работы [4]:

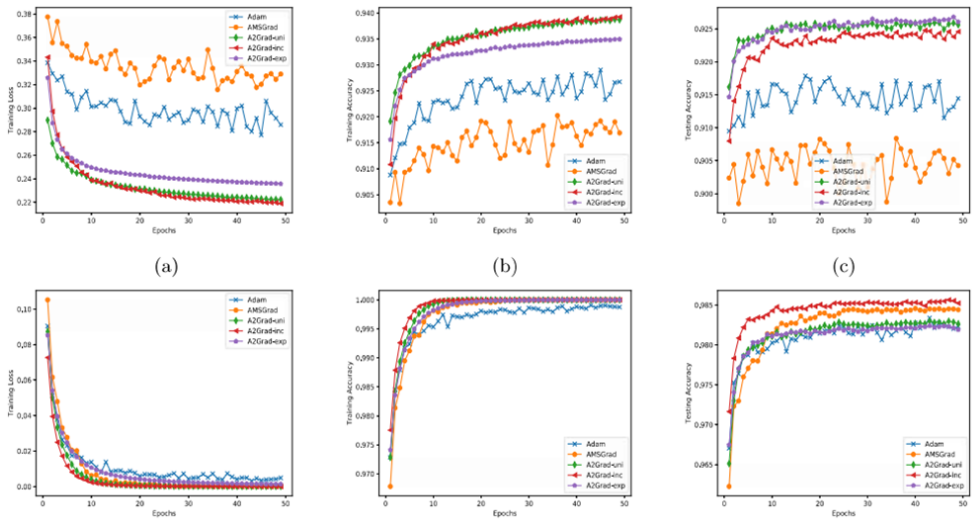

Figure 1: Comparison of algorithm performance on convex and nonconvex models. From left to right, each column represents the training loss, training accuracy and test accuracy. The first and second row plot the experimental results for logistic regression and neural network on the MNIST dataset respectively. The third and fourth rows plot the results for Cifarnet and Vgg16 on CIFAR10 respectively

3 Литературный обзор

1. **А.В. Гасников. Современные численные методы оптимизации. Метод универсального градиентного спуска. 2018 [1]**

   Это основной источник, в котором адаптивный стохастический градиентный спуск рассматривается как модификация универсального градиентного спуска, приведены оценки на скорость сходимости для стохастического градиента с известными параметрами L, D для выпуклых и сильно выпуклых функций. Также в [1] описаны преимущества концепции минибатчинга в задачах минимизации суммы функций.

   В данном пособии показано, что оптимальные оценки для детерминированных методов переносятся на стохастический случай, однако это может быть не выполнено для адаптивных методов. В частности, хотелось бы перенести качество работы универсального градиентного спуска на стохастический случай, но этого сделать не удаётся.

   Похожие рассуждения можно провести и для зеркального спуска, реализовав его стохастический вариант, однако это выходит за рамки программы минимум данного проекта.

2. **Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. 2009 [3]**

   В работе показано, что оценки, приведённые в [1], являются неулучшаемыми на соответствующих классах. При этом, как показано в [1], они достигаются на неоптимальных для детерминированного случая методах!

3. **John C Duchi. Introductory lectures on stochastic optimization. 2017 [5]** Это пособие написано автором метода AdaGrad, в своей работе он приводит одну из часто рассматриваемых

вариаций метода:

$$x_{k+1} = \underset{x \in Q}{\operatorname{argmin}} \left\{ \langle g_k, x \rangle + \frac{1}{2} \langle x, H_k x \rangle \right\} = \underset{x \in Q}{\operatorname{argmin}} \left\{ \|x - (x_k - H_k^{-1} g_k)\|_{H_k}^2 \right\}, \tag{8}$$

$$H_k = \frac{1}{\alpha} diag \left(\sum_{i=1}^{k} g_i g_i^\top \right)^{1/2}, \tag{9}$$

$$\mathbb{E}[g_k | x_k] \in \partial f(x_k).$$

Хотя этот метод не является оптимальным в общем случае, при некоторых начальных условиях гарантировано превосходит по качеству обычный стохастический градиентный спуск. Также метод AdaGrad хорошо работает с разреженными градиентами, о чём можно сделать вывод по приведённым в работе оценкам сходимости. Основные оценки скорости сходимости приведены в [5].

4. **Diederik P. Kingma, Jimmy Lei Ba. Adam: a method for stochastic optimization. 2014 [6]**

   В качестве метода, с которым стоит сравнивать качество работы реализуемого метода обязательно должен быть выбран и метод Adam, стремительно завоевавший мировую популярность. Метод основывается на адаптивном вычислении первого и второго моментов. Привлекательность метода - в том что он требует малых вычислительных затрат и количества памяти. Более того, величина подбираемых параметров инвариантна к масштабированию градиента, метод также хорошо работает с разреженными градиентами.

Algorithm 2 Adam. g_t^2 означает поэлементное произведение $g_t \odot g_t$. Все векторные операции - поэлементные.

Input: α : Stepsize **Input**: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates **Input**: θ_0 : Initial parameter vector

$m_0 = 0$ (Initialize 1^{st} moment vector)
$v_0 = 0$ (Initialize 2^{nd} moment vector)
$t = 0$ (Initialize timestep)
while θ_t not converged **do**
  $t = t + 1$
  generate $\{\xi^l\}_{l=1}^r$
  $g_t = \nabla_\theta f(\theta_t, \{\xi^l\}_{l=1}^r)$
  $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
  $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
  $\hat{m}_t = m_t / (1 - \beta_1^t)$
  $\hat{v}_t = v_t / (1 - \beta_2^t)$
  $\theta_t = \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t + \varepsilon})$
end while

Output: θ_t

В работе также приведены оценки на скорость сходимости. В качестве следствия из теоремы о сходимости метода присутствует утверждение, что отношение суммарной невязки к количеству функций, сумма которых минимизируется, стремится к нулю при увеличении количества функций. Это приятное следствие наталкивает на решение использовать это метод во многих задачах машинного обучения.

5. **Qi Deng, Yi Cheng, and Guanghui Lan. Optimal adaptive and accelerated stochastic gradient descent. 2018 [4]**

   В этой статье авторы уделили внимание различным вариациям метода AdaGrad. Самая базовая реализация может быть записана следующим образом:

Algorithm 3 AdaGrad

Input: $x_0,\ v_{-1}$

1: **for** $k = 0, 1, ..., K$ **do**
2: Generate ξ_k
3: Compute $G_k \in \nabla f(x_k, \xi_k)$
4: Set $v_k = v_{k-1} + G_k^2$
5: $x_{k+1} = x_k - \beta_k G_k / \sqrt{v_k}$
6: **end for**

Output: x_{K+1}

В статье приведена также намного более общая конструкция в терминах прокс-функций и дивергенции Брэгмана:

Algorithm 4 Adaptive accelerated stochastic gradient (A2Grad) algorithm

Input: $x_0,\ \overline{x_0},\ \gamma_k,\ \beta_k > 0$

1: **for** $k = 0, 1, ..., K$ **do**
2: Update

$$\underline{x_k} = (1 - \alpha_k)\overline{x_k} + \alpha_k x_k$$

3: Sample ξ_k
4: Compute $\underline{G_k} \in \nabla f(\underline{x_k}, \xi_k)$
5: Compute $\phi_k(.)$
6: Update

$$x_{k+1} = \underset{x \in Q}{\operatorname{argmin}} \langle \underline{G_k}, x \rangle + \gamma_k D(x_k, x) + \beta_k D_{\phi_k}(x_k, x)$$

$$\overline{x_{k+1}} = (1 - \alpha_k)\overline{x_k} + \alpha_k x_{k+1}$$

7: **end for**

Output: $\overline{x_K}$

$D(x, y) \equiv D_\psi(x, y)$ - дивергенция Брэгмана, где $\psi(x)$ - произвольная заранее выбранная выпуклая прокс-функция (в работе [4] используется обычная Евклидова прокс-функция $\frac{1}{2}\|x\|^2$).

$\phi(x)_k$ - прокс-функция, адаптивная по последнему посчитанному градиенту.

В работе [4] приведено целое семейство state-of-the-art вариаций метода AdaGrad, отличающихся выбором адаптивности прокс-функции ϕ_k.

Рисунок 1 отображает результаты экспериментов по сравнению различных имплементаций адаптивного стохастического градиентного спуска ([4]).

Видно, насколько в некоторых случаях предложенные вариации превосходят в качестве и стандартный метод AdaGrad и описанный выше метод Adam.

6. **Yehuda KLr Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. 2018 [7]**

В работе доказана теорема о сходимости классического AdaGrad. При этом на функцию накладываются сильные ограничения: выпуклость и β-гладкость. Тогда, если оракул выдаёт зашумлённый градиент, то справедлив следующий результат:

$$\mathbb{E}(f(\overline{x}_N)) - \min_{x \in \mathbb{R}^d} f(x) \leq O\left(\frac{\beta R^2}{T} + \frac{DR}{\sqrt{T}}\right) \tag{10}$$

7. **J. Wright Stephen. Optimization algorithms for data analysis. 2016 [8]**

В пособии описаны классические задачи машинного обучения в терминах оптимизации функций потерь.

Например, для алгоритма логистической регрессии минимизируется минус логарифм функции правдоподоб

$$L(X) := \frac{1}{m}\sum_{j=1}^{m}\left[\sum_{l=1}^{M} y_{jl}(x_{(l)}^{\top}a_j) - \log\left(\sum_{l=1}^{M}\exp(x_{(l)}^{\top}a_j)\right)\right]. \tag{11}$$

8. **А.В. Гасников, П.Е. Двуреченский, Ю.Е. Нестеров. Стохастические градиентные методы с неточным оракулом. 2016 [2]**

   В работе приведён обзор метода рандомизации суммы и приводятся оценки для негладких, гладких выпуклых и сильно выпуклых функций. Приведённые оценки сходимости, правда, отличаются от нижних оценок сходимости для детерминированных методов. Кроме того, ещё раз было показано, как с помощью регуляризации оценки для сильно выпуклого случаю переносятся на выпуклый.

9. **Sebastian Ruder. An overview of gradient descent optimization algorithms. 2016 [9]**

   В этой популярной статье производится краткий обзор градиентных методов с точки зрения их реализации и соответствия различным задачам. В частности в статье уделено внимание градиентному спуску с минибатчингом, указаны основные нюансы, на которые стоит обращать внимание на практике. Стоит отметить, что под минибатчингом в статье понимается, что на одном шаге градиент считается по набору последовательных элементов тренировочной выборки, т.е. как это обычно бывает на практике. Автор также высказывает предположение о преимуществах минибатчинга при тренировке нейросетей, связывая это с "сильной негладкостью" минимизируемого функционала и, как следствие, наличие большого числа локальных оптимумов.

4 Метрики качества

Так как проект заключается в том, чтобы проверить качество работы алгоритма с помощью различных численных экспериментов, все метрики качества были описаны в разделе **Outcomes**.

5 Примерный план

- Сначала будет реализован метод.
- Потом его качество будет проверено на различных задачах, описанных выше (задачи машинного обучения, нейронные сети на датасетах `MNIST`, `CIFAR10`) в сравнении с другими популярными адаптивными методами.
- Также стоит задача проверить качество алгоритма на более узких классах задач.
- Можно также проверить работу метода на более сложных моделях (глубоких нейросетях, конволюционных нейросетях). Кроме того можно реализвовать стохастический зеркальный спуск.

References

[1] АВ Гасников. Современные численные методы оптимизации. Метод универсального градиентного спуска. *М.: МФТИ*, 2018.

[2] АВ Гасников, ПЕ Двуреченский, and ЮЕ Нестеров. Стохастические градиентные методы с неточным оракулом. *Труды МФТИ*, 8(1):41–91, 2016.

[3] Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

[4] Qi Deng, Yi Cheng, and Guanghui Lan. Optimal adaptive and accelerated stochastic gradient descent. *arXiv preprint arXiv:1810.00553*, 2018.

[5] John C Duchi. Introductory lectures on stochastic optimization. *Mathematics of Data*, 16:1455, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Yehuda Kfir Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. In *Advances in Neural Information Processing Systems*, pages 6501–6510, 2018.

[8] J. Wright Stephen. Optimization algorithms for data analysis. *IAS/Park City Mathematics Series*, 2016.

[9] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.