Stochastic Quasi Newton methods for Neural Networks

Yakovlev Konstantin iakovlev.kd@phystech.edu
Daniil Merkulov daniil.merkulov@skoltech.ru

Project Proposal

В проекте рассматривается стохастический квазиньютоновский метод. Данный проект существенно опирается на работу [1]. Приведенный здесь алгоритм использует классическую формулу обновления BFGS в ее ограниченной форме. Также он является эффективным, робастным, масштабируемым и имеет обещающие перспективы в машинном обучении. В данной работе предлагается реализовать приведенный алгоритм и сравнить его с алгоритмом SGD на примере обучения нейронной сети.

1 Идея

Прямое применение классических квазиньютоновских методов обновления для детерминированной оптимизации приводит к шумным текущим оценкам, которые оказывают негативное влияние на устойчивость итерации. Идея заключается в том, чтобы использовать стохастический квазиньютоновский метод, который сможет преодолеть данную проблему.

1.1 Problem

Пусть задана некоторая функция $f(w, \xi)$, где $w \in \mathbb{R}^n$, ξ – случайный вектор (x, z). Можно смотреть на пару (x, z) как на объект и ответ соответственно. Обычно в машинном обучении функция $f(w, \xi)$ имеет вид:

$$f(w, \xi) = \ell(h(w; x_i); z_i), \tag{1}$$

где ℓ – неотрицательная функция потерь, h – модель, параметризованная вектором w, $\{x_i, z_i\}_{i=1}^N$ – обучающая выборка. Определим эмпирический риск как:

$$F(w) = \frac{1}{N} \sum_{i=1}^{N} f(w; x_i, z_i) \tag{2}$$

Определим множество $\mathcal{S} \subset \{1, \ldots, N\}$. Пусть $b = |\mathcal{S}| \ll N$. Тогда можно записать оценку для эмпирического риска:

$$\widehat{\nabla} F(w) = \frac{1}{b} \sum_{i \in \mathcal{S}} \nabla f(w; x_i, z_i) \tag{3}$$

Сформулируем задачу оптимизации следующим образом:

$$\min_{w \in \mathbb{R}^n} F(w) \tag{4}$$

TODO:...Дальнейшее описание:

$$\widehat{\nabla}^2 F(w) = \frac{1}{b_H} \sum_{i \in \mathcal{S}_H} \nabla^2 f(w; x_i, z_i) \tag{5}$$

Запишем итоговый алгоритм:
Запишем алгоритм обновления гессиана:

Algorithm 1 Stochastic Quasi-Newton Method (SQN)

Input: начальные параметры w^1; натуральные числа M, L; последовательность шагов $\{\alpha_k\}$

Инициализируем $t = -1$
Инициализируем $\overline{w}_t = 0$
for $k = 1, 2, ...,$ **do**
  Выберем множество $\mathcal{S} \subset \{1, \ldots, N\}$
  Вычислим $\widehat{\nabla} F(w^k)$ по формуле 3
  $\overline{w}_t = \overline{w}_t + w^k$
  if $k \leq 2L$ **then**
    $w^{k+1} = w^k - \alpha_k \widehat{\nabla} F(w^k)$
  else
    $w^{k+1} = w^k - \alpha_k H_t \widehat{\nabla} F(w^k)$, H_t оперделяется алгоритмом 2
  end if
  if $k \ \% \ L = 0$ **then**
    $t = t + 1$
    $\overline{w}_t = \overline{w}_t / L$
    if $t > 0$ **then**
      Выбираем $\mathcal{S}_H \subset \{1, \ldots, N\}$, чтобы определить $\widehat{\nabla}^2 F(\overline{w}_t)$ по формуле 5
      Вычислим $s_t = \overline{w}_t - \overline{w}_{t-1}$, $y_t = \widehat{\nabla}^2 F(\overline{w}_t)(\overline{w}_t - \overline{w}_{t-1})$
    end if
    $\overline{w}_t = 0$
  end if
end for

Algorithm 2 Обновление Гессиана

Input: Счетчик обновлений t, натуральное число M , коррекционные пары (s_j, y_j), $j = t - \min(t, M) + 1, \ldots, t$
Output: H_t

Инициализировать $H = (s_t^\top y_t)/(y_t^\top y_t) I$, где s_t, y_t вычислены на шаге 17 алгоритма 1
for $j = t - \min(t, M) + 1, \ldots, t$ **do**
  $\rho_j = 1 / y_j^\top s_j$
  Применяем BFGS формулу $H_t = (I - \rho_j s_j y_j^\top) H (I - \rho_j y_j s_j^\top) + \rho_j s_j s_j^\top$
end for
return H_t

2 Outcomes

Результатами данного проекта являются:

- Реализация алгоритма SQN, описанного в работе [1], с использованием фреймворка jax [2].
- Сравнение алгоитма SQN [1] с алгоритмом SGD на примере обучения нейронной сети. Метрики качества приведены далее.

3 Литературный обзор

В последнее время в машинном обучении растет интерес к очень большим моделям. В большинстве крупномасштабных задач обучения таких моделей используются алгоритмы стохастической оптимизации, которые обновляют параметры модели на основе небольшого количества обучающих данных. Приведем обзор некоторых работ по стохастической оптимизации.

В работе [1] рассматривалась задача минимизации выпуклой стохастической функции. Предложен квазиньютоновский метод, испольщующий формулу обновления BFGS с ограниченной памятью. Здесь приведен метод получения стабильных оценок гессиана. Идея состоит в том, чтобы вычислить средние оценки кривизны через регулярные промежутки времени с помощью некоторой подвыборки.

Также рассматривалась задача минимизации невыпуклой стохастической функции [3]. В данной работе представлена общая структура стохастических квази-ньютоновских методов невыпуклой стохастической

оптимизации. Также был предложен алгоритм SdLBFGS (stochastic damped L-BFGS), соответствующий данной структуре. Были приведены и доказаны достаточные условия сходимости алгоритма. Кроме того, существует улучшение данного алгоритма – Sd-REG-LBGS [4]. Эксперименты показывают, что новая версия алгоритма в целом превосходит SdLBFGS. Также для этого метода получены достаточные условия сходимости.

В некоторых работах рассматривалась задача минимизации сильно выпуклой функции на римановских многообразиях [5]. Данный подход является нестандартным, поскольку подобные алгоритмы хорошо изучены лишь в случае евклидовых пространств. Для решения данной задачи в работе был предложен алгоритм Riemannian Stochastic VR L-BFGS. Вычислительные эксперименты показали, что построенный алгоритм превосходит другие алгоритмы римановой оптимизации, а также работает значительно лучше, чем один из самых быстрых стохастических алгоритмов в евклидовом пространстве, VR-PCA [6]. Алгоритм VR-PCA использует технику уменьшения стохастической градиентной дисперсии [7]. Стоит также отметить, что алгоритм имеет линейную сходимость к оптимальному решению.

Рассматривались также подходы с использованием метода ускоренного градиента Нестерова [8]. В данной работе рассматривалась крупномасштабная задача минимизации невыпуклой стохастической функции. К квазиньютоновскому стохастическому методу были добавлены некоторые модификации для ускорения сходимости. Вычислительные эксперименты показывают, что предлагаемые метод oNAQ (online NAQ) превосходит традиционные методы oBFGS (online BFGS) [9] с одинаковыми затратами на вычисление и память.

Также был предложен новый класс стохастических адаптивных методов минимизации самосогласованных стохастических функций [10]. Ключевой идеей адаптивных методов является то, что шаг t_k на каждой итерации может быть вычислен аналитически, используя только локальную информацию. В частности, методы этого класса включают в себя модификации градиентного спуска (GD) и BFGS. Также в работе [11] было показано, что метод BFGS с адаптивным шагом имеет сверхлинейную сходимость для сильно выпуклых самосогласованных функций.

4 Метрики качества

Метрики для второй части моего проекта – сравнение SGD и SQN:

- Значение оптимизируемой функции после одинакового времени работы на CPU/GPU.
- Значение оптимизируемой функции после одинакового количества итераций.
- Количество вызовов оптимизируемой функции и ее градиента после одинакового количества итераций.

5 Примерный план

1. Первым делом реализуем SQN с использованием фреймворка JAX [2]. По сути, в нашем проекте мы представили псевдокод. Срок: 21 марта.
2. Затем займемся планированием эксперимента для сравнения SGD и SQN. В качестве архитектуры нейронной сети можно взять LeNet, а в качестве выборки – MNIST. Нейронная сеть будет реализована на PyTorch. В качестве оптимизатора будет подаваться SGD и SQN. Срок: 31 марта.

References

[1] Richard H. Byrd, S. L. Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM J. Optim*, 26(2):1008–1031, 2016.

[2] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018.

[3] Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu 0005. Stochastic quasi-newton methods for nonconvex stochastic optimization. *SIAM J. Optim*, 27(2):927–956, 2017.

[4] H. Chen, H. C. Wu 0001, S. C. Chan, and Wong-Hing Lam. A stochastic quasi-newton method for large-scale nonconvex optimization with applications. *CoRR*, abs/1912.04456, 2019.

[5] Anirban Roychowdhury and Srinivasan Parthasarathy 0001. Accelerated stochastic quasi-newton optimization on riemann manifolds. *CoRR*, abs/1704.01700, 2017.

[6] Ohad Shamir. A stochastic pca algorithm with an exponential convergence rate. *CoRR*, abs/1409.2848, 2014.

[7] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. 26:1–9, 2013.

[8] S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. A stochastic quasi-newton method with nesterov's accelerated gradient. *CoRR*, abs/1909.03621, 2019.

[9] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, March 21-24, 2007*, volume 2 of *JMLR Proceedings*, pages 436–443. JMLR.org, 2007.

[10] Chaoxu Zhou, Wenbo Gao, and Donald Goldfarb. Stochastic adaptive quasi-newton methods for minimizing expected values. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 4150–4159. PMLR, 2017.

[11] Wenbo Gao and Donald Goldfarb. Quasi-newton methods: superlinear convergence without line searches for self-concordant functions. *Optim. Methods Softw*, 34(1):194–217, 2019.