Алгоритм Natasha-2

2 мая 2020 г.

1 Постановка задачи

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

где f_i — в общем случае невыпуклые функции.

Будет построен онлайн-алгоритм (т.е. не зависящий от n), находящий локальный минимум с заданной точностью.

2 Необходимые предположения

- f имеет ограниченную дисперсию, откуда $\mathbb{E}_i \left[\|\nabla f_i(x) - \nabla f(x)\| \right]$
- L-Липшицевость градиента: $\|\nabla f(x) - \nabla f(y)\| \leq L \cdot \|x - y\|$
- L_2-Липшицевость гессиана: $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L_2 \cdot \|x - y\|$

3 Определения

- Локальный минимум с (ε, δ)-точностью - точка x, такая что $\|\nabla f(x)\| < \varepsilon$ и $\nabla^2 f(x) \succeq -\delta \mathbf{I}$ (все собственные значения гессиана больше $-\delta$).
- σ-невыпуклая функция - функция, такая что минимальное собственное значение её гессиана превосходит $-\sigma$

4 Описание алгоритма

Главная идея - использование гессиана для избегания седловых точек.

4.1 Carmon: Сведение к поиску стационарных точек

На каждой итерации в точке y_k сравниваем минимальное собственное значение $\lambda_{min}(\nabla^2 f(y_k))$ и $-\delta$.

- Если $\exists v : v^T \nabla^2 f(y_k) v \leq -\frac{\delta}{2}$, то пытаемся отдалиться от седла: $y_{k+1} = y_k \pm \frac{\delta}{L_2} v$ ($+$ или $-$ выбираем случайно)
- Иначе берём $F^k(x) = f(x) + L(\max\{0, \|x - y_k\| - \frac{\delta}{L_2}\})^2$, которая являлется $5L$-Липшицевой по градиенту и 3δ-невыпуклой; ищем для неё точку $\widehat{x} : \|\nabla F^k(\widehat{x})\| \leq \varepsilon$ и переходим туда: $y_{k+1} = \widehat{x}$. Как видно, F^k штрафует за выход из «безопасной зоны» $\{x : \|x - y_k\| \leq \frac{\delta}{L_2}\}$

4.2 Natasha 1.5: Поиск стационарных точек для σ-невыпуклых функций

Каждая эпоха делится на $p = \Theta((\frac{\sigma}{\varepsilon L})^{\frac{2}{3}})$ суб-эпох со стартовым вектором $\widehat{x}$.

Затем, при подсчёте градиентов, вместо $f(x)$ фактически используется $f(x) + \sigma\|x - \widehat{x}\|^2$, что должно стабилизировать алгоритм, немного замедляя его. Вместе с этим, используется пересчёт градиента из алгоритма SVRG

Псевдокод:

Algorithm 1 $\texttt{Natasha1.5}(F, x^{\varnothing}, B, T', \alpha)$

Input: $f(\cdot) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, starting vector $x^{\varnothing}$, epoch length $B \in [n]$, epoch count $T' \geq 1$, learning rate $\alpha > 0$.

$\widehat{\mathsf{x}} \leftarrow x^{\varnothing}$; $\quad p \leftarrow \Theta((\sigma/\varepsilon L)^{2/3})$; $\quad m \leftarrow B/p$; $\quad X \leftarrow []$;

for $k \leftarrow 1$ **to** T' **do** ⋄ *T' epochs each of length B*

  $\widetilde{\mathbf{x}} \leftarrow \widehat{\mathsf{x}}$; $\mu \leftarrow \frac{1}{B}\sum_{i \in S} \nabla f_i(\widetilde{\mathbf{x}})$ where S is a uniform random subset of $[n]$ with $|S| = B$;

  for $s \leftarrow 0$ **to** $p-1$ **do** ⋄ *p sub-epochs each of length m*

    $x_0 \leftarrow \widehat{\mathsf{x}}$; $\quad X \leftarrow [X, \widehat{\mathsf{x}}]$;

    for $t \leftarrow 0$ **to** $m-1$ **do**

      $\widetilde{\nabla} \leftarrow \nabla f_i(x_t) - \nabla f_i(\widetilde{\mathbf{x}}) + \mu + 2\sigma(x_t - \widehat{\mathsf{x}})$ where $i \in_R [n]$

      $x_{t+1} = x_t - \alpha \widetilde{\nabla}$;

    end for

    $\widehat{\mathsf{x}} \leftarrow$ a random choice from $\{x_0, x_1, \ldots, x_{m-1}\}$; ⋄ *in practice, choose the average*

  end for

end for

$\widehat{\mathsf{y}} \leftarrow$ a random vector in X. ⋄ *in practice, simply return $\widehat{\mathsf{y}}$*

4.3 Oja: Быстрый поиск максимального собственного значения