Применения операторов сжатия в задачах распределенной оптимизации

Касюк В.А.

Московский физико-технический институт

Москва,
2024 г.

Цели исследования

Целью исследования является создание новых, более совершенных операторов сжатия для задач распределенной оптимизации, а также анализ их особенностей.

Проблематика данной задачи заключается в том, что предыдущие исследования показывают эмпирически лучшую сходимость смещенных операторов сжатия, однако теория мало развита и построено мало методов.

О смещенном сжатии для распределенного обучения

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, Mher Safaryan On Biased Compression for Distributed Learning

Унифицированная теория методов SGD

Eduard Gorbunov, Filip Hanzely, Peter Richtárik A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent

Дополнительные ссылки

Е.А Воронцова, Р. Хильдебранд, А.В. Гасников, Ф.С. Стонякин Выпуклая оптимизация

Постановка задачи

Решается задача оптимизации :

$$\min_{x \in R^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x), \qquad (0.1)$$

где $x \in R^d$ - пространство признаков размерности d, n - количество устройст/узлов, $f_i(x) : R^d \to R$ - потери, которые получает модель на устройстве под номером i. Часто функция потерь имеем вид :

$$f_i(x) = E_{\xi \sim P_i}[f_\xi(x)],$$

где P_i - это распределение тренировочных данных на устройстве i. То, как данные распределяются между работниками оказывает большое влияние на процесс обучения.

Предыдущие результаты: методы Top-k, Rand-K

Базовым решением задачи (0.1) является обычный градиентный спуск (GD), который имеет вид :

$$x^{k+1} = x^k - \frac{\eta^k}{n}\sum_{i=1}^{n}\nabla f_i(x^k) \tag{0.2}$$

Теперь поставим задачу оптимизаци, которую будем решать с помощью градиентного спуска с сжатием(CGD - compressed gredient descent). Пускай решается задача:

$$\min_{x\in R^d} f(x) \tag{0.3}$$

, где $f : R^d \to R$, L - гладкая и μ-сильно-выпуклая, методом :

$$x^{k+1} = x^k - \eta C(\nabla f(x)) \tag{0.4}$$

, где $C : R^d \to R^d$ - оператор сжатия и $\eta > 0$ - темп обучения.

Предыдущие результаты: методы Top-k, Rand-K

Под операторами сжатия мы подразумеваем (возможно рандомизированное) отображение $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ с некоторыми требованиями. Как правило, в литературе рассматриваются *несмещенные* операторы сжатия $\mathcal{C}$ с ограниченным вторым моментом, т.е. :

Definition

Пусть $\zeta \geq 1$. Тогда $\mathcal{C} \in \mathbb{U}(\zeta)$ если $\mathcal{C}$ явлется несмещенным (то есть, $\mathrm{E}\left[\mathcal{C}(x)\right] = x$ для любых x) и втором момент ограничен [a]

$$\mathrm{E}\left[\|\mathcal{C}(x)\|_2^2\right] \leq \zeta \|x\|_2^2, \qquad \forall x \in \mathbb{R}^d. \tag{0.5}$$

[a] (0.5) Что также может быть переписанное в виде $\mathrm{E}\left[\|\mathcal{C}(x) - x\|_2^2\right] \leq (\zeta - 1)\|x\|_2^2$.

Предыдущие результаты: методы Top-k, Rand-K

Definition

Будем говорить, что $\mathcal{C} \in \mathbb{B}^1(\alpha, \beta)$ для некторых $\alpha, \beta > 0$ если

$$\alpha \|x\|_2^2 \leq \mathrm{E}\left[\|\mathcal{C}(x)\|_2^2\right] \leq \beta \langle \mathrm{E}\left[\mathcal{C}(x)\right], x\rangle, \qquad \forall x \in \mathbb{R}^d \,. \tag{0.6}$$

Definition

Будем говорить, что $\mathcal{C} \in \mathbb{B}^2(\gamma, \beta)$ для некторых $\gamma, \beta > 0$ если

$$\max\left\{\gamma \|x\|_2^2, \tfrac{1}{\beta}\mathrm{E}\left[\|\mathcal{C}(x)\|_2^2\right]\right\} \leq \langle \mathrm{E}\left[\mathcal{C}(x)\right], x\rangle \qquad \forall x \in \mathbb{R}^d \,. \tag{0.7}$$

Definition

Будем говорить, что $\mathcal{C} \in \mathbb{B}^3(\delta)$ для некоторого $\delta > 0$ если

$$\mathrm{E}\left[\|\mathcal{C}(x) - x\|_2^2\right] \leq \left(1 - \frac{1}{\delta}\right) \|x\|_2^2, \qquad \forall x \in \mathbb{R}^d \,. \tag{0.8}$$

Для $k \in [d] \stackrel{\text{def}}{=} \{1, \ldots, d\}$, **несмещенный случайный (aka Rand-k) сжимающий** определяется как :

$$\mathcal{C}(x) \stackrel{\text{def}}{=} \frac{d}{k} \sum_{i \in S} x_i e_i, \tag{0.9}$$

где $S \subseteq [d]$ подмножество индексов размера k выбранных равновероятно, и $e_1, \ldots, e_d$ стандартные базисные вектора $\mathbb{R}^d$.

Lemma

Rand-k оператор (0.9) принадлежит $U(\frac{d}{k})$.

Жадный (aka Top-k) сжимающий оператор определяется, как :

$$\mathcal{C}(x) \overset{\text{def}}{=} \sum_{i=d-k+1}^{d} x_{(i)} e_{(i)}, \tag{0.10}$$

где координаты упорядочены по их абсолютным значениям $|x_{(1)}| \le |x_{(2)}| \le \dots \le |x_{(d)}|$.

Lemma

Top-k сжимающий оператор (0.10) принадлежит $\mathbb{B}^1(\frac{k}{d}, 1)$, $\mathbb{B}^2(\frac{k}{d}, 1)$, *и* $\mathbb{B}^3(\frac{d}{k})$.

Предлагается рассмотреть 3 новый оператора сжатия вида :

$$C(\nabla f(x)) = Top_k(\nabla f(x)|w) \tag{0.11}$$

, где w имеет смысл вектора "важности", а $Top_k(\circ|w)$ - выбор самых важных координат(обладающих самыми большими весами "важности" - координатами w_i). В зависимости от того, как определеяется вектор w, меняется и сам оператор.

(a) $w_i := f\left(x^k\right) - f\left(x^k - \eta\left[\nabla f(x^k)\right]_i\right).$

(b) $w := \arg\min\limits_{w \in \Delta} f\left(x^k - \eta \sum_{i=1}^d w_i \left[\nabla f(x^k)\right]_i\right).$

(c) $w := \arg\min\limits_{w \in [0,1]^d} f\left(x^k - \eta \sum_{i=1}^d w_i \left[\nabla f(x^k)\right]_i\right)$

Рассмотрим работу новых операторов на примере задачи логистической регрессии классическим методом и с использованием нейросетей. Следует обратить внимание, что на каждом шаге алгоритмов (b) и (c) необходимо решать свою задачи оптимизации. Конкретно для случая (b) воспользуемся методом зеркального спуска.

На каждой итерации спуска нам нужно решать задачу:

$$\min_{w \in \Delta^d} g_k(w),$$

где

$$g_k(w) := f\left(x^k - \eta\, w^T \cdot \mathcal{P}\right),$$

$$\mathcal{P} := \mathcal{I} \odot \nabla f\left(x^k\right) = \left(\left[\nabla f\left(x^k\right)\right]_1, \left[\nabla f\left(x^k\right)\right]_2, \ldots, \left[\nabla f\left(x^k\right)\right]_d\right)$$

Решать ее будем методом зеркального спуска с KL-дивергенцией.

Решение:

$$w_i^{k+1} := \frac{w_i^k e^{-\eta\left[\nabla g(w^k)\right]_i}}{\sum\limits_{j=1}^{d} w_j^k e^{-\eta\left[\nabla g(w^k)\right]_j}}$$

Здесь :

$$\nabla g(w) = -\eta\, \mathcal{P} \cdot \nabla f\left(x^k - \eta\, w^T \cdot \mathcal{P}\right)$$

Классическая логистическая регрессия :

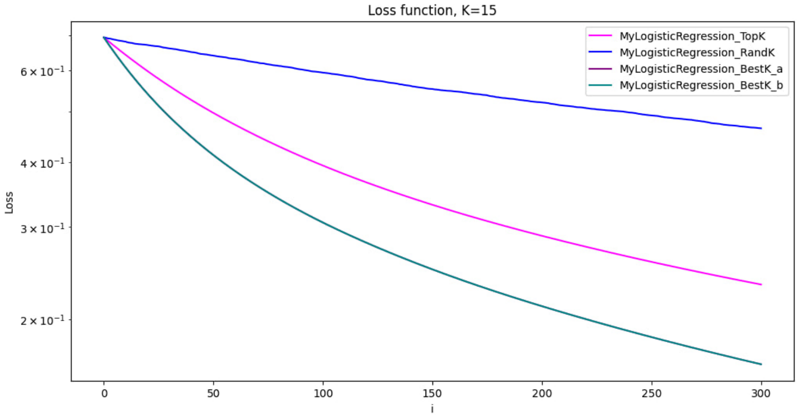

Рис.: Зависимость функции потерь от итерации

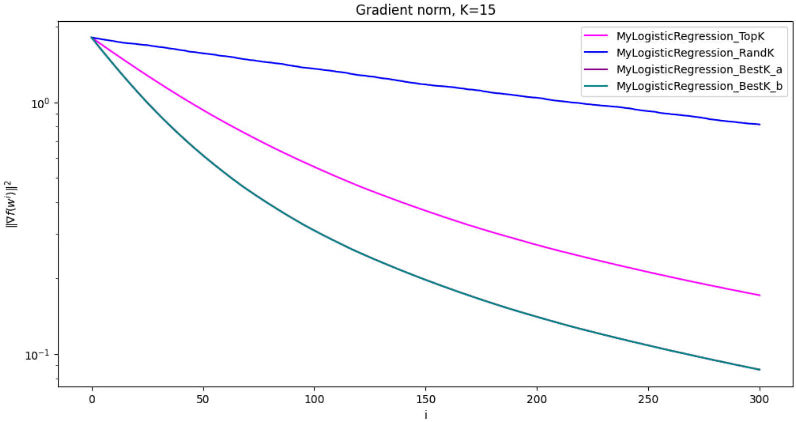

Рис.: Зависимость нормы градиента от итерации

Теперь будем обучать нейросеть для датасета mushrooms. Получаем следующие результаты :

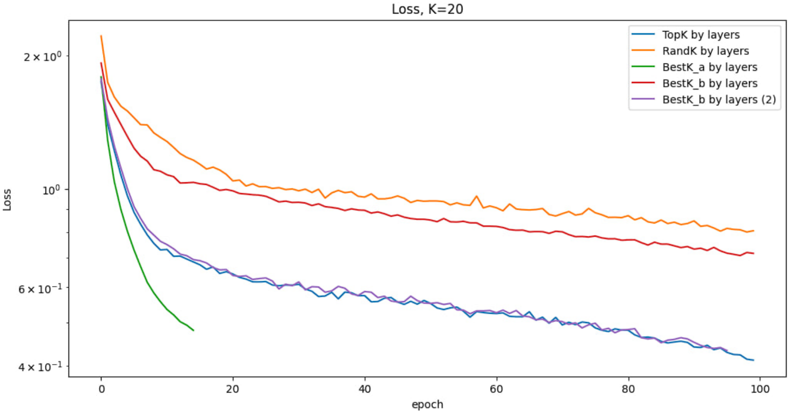

Рис.: Зависимость функции потерь от эпохи

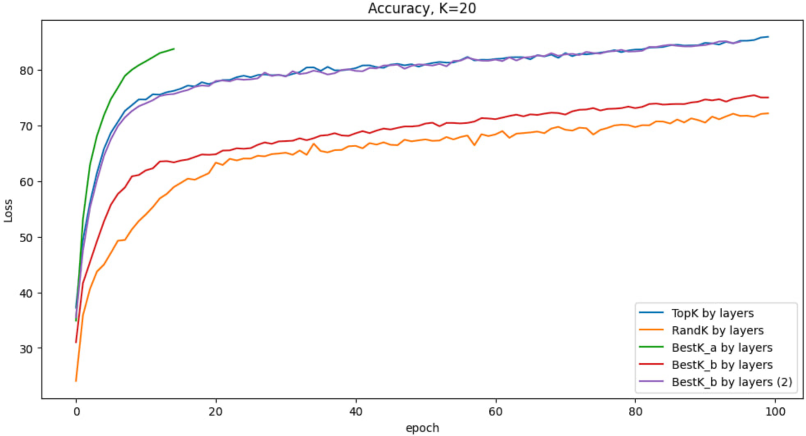

Рис.: Зависимость точности от эпохи

Хочу обратить внимание на реальное время для обучения :

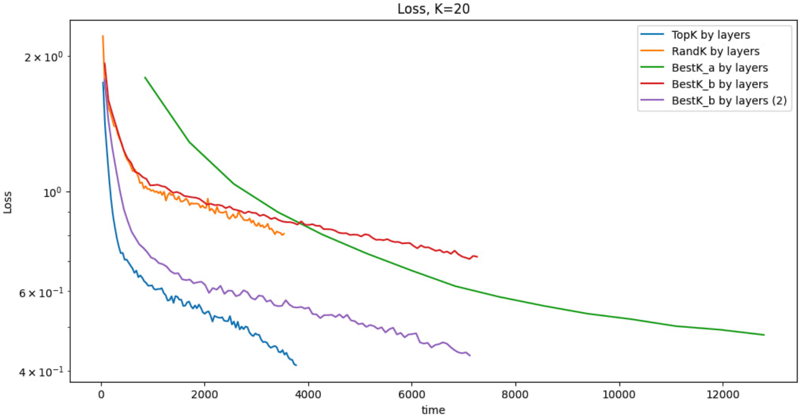

Рис.: Зависимость функции потерь от времени обучения

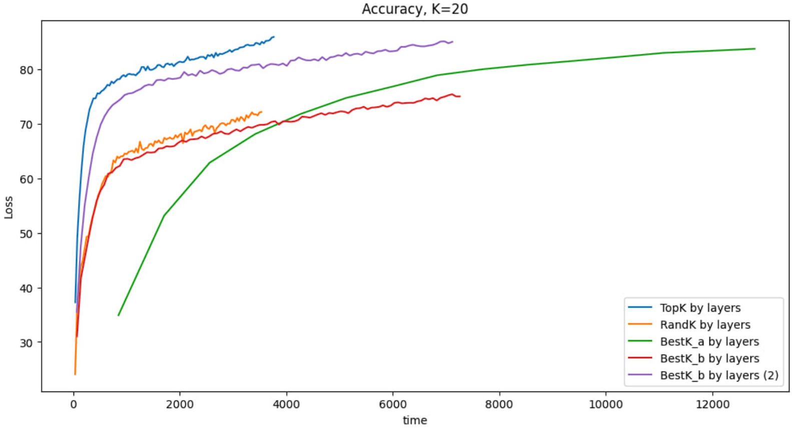

Рис.: Зависимость точности от времени

Анализ полученных результатов

Можно сделать следующие выводы относительно результатов эксперимента

1° Методы действительно сходятся.

2° Методы показывают скорость сходимости более высокую, чем $Rand_k$, что согласуется с теорией.

3° Имеет место более высокая скорость сходимости методов (a), (b) и (c) в некоторых задачах. Однако они обходят вычислительно дороже, чем стандартные.

Подводя итоги работы можно заключить, что :

1° Предложены и опробованы 3 новых оператора сжатия

2° Получены гарантии сходимости этих методов

3° Показано превосходство новых методов над стандартными.