

Ускорения градиентного спуска

Семинар

Оптимизация для всех! ЦУ

Скорости сходимости градиентного спуска

$$\min_{x \in \mathbb{R}^n} f(x) \quad x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad \kappa = \frac{L}{\mu}$$

| | выпуклая и гладкая | выпуклая и сильно выпуклая (или PL) |
|----------------|--|---|
| Верхняя оценка | $f(x_k) - f^* \approx \mathcal{O}\left(\frac{1}{k}\right)$ | $\ x_k - x^*\ ^2 \approx \mathcal{O}\left(\left(\frac{\kappa - 1}{\kappa + 1}\right)^k\right)$ |
| Нижняя оценка | $f(x_k) - f^* \approx \Omega\left(\frac{1}{k^2}\right)$ | $\ x_k - x^*\ ^2 \approx \Omega\left(\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k\right)$ |

Три схемы обновления

- Градиентный спуск

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

Перемещаем точку x_k в направлении $-\nabla f(x_k)$ на $\alpha_k \|\nabla f(x_k)\|$ единиц.

Три схемы обновления

- **Градиентный спуск**

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

Перемещаем точку x_k в направлении $-\nabla f(x_k)$ на $\alpha_k \|\nabla f(x_k)\|$ единиц.

- **Метод тяжелого шарика**

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

Выполняем GD, перемещаем обновленный x в направлении предыдущего шага на $\beta_k \|x_k - x_{k-1}\|$ единиц.

Три схемы обновления

- Градиентный спуск

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

Перемещаем точку x_k в направлении $-\nabla f(x_k)$ на $\alpha_k \|\nabla f(x_k)\|$ единиц.

- Метод тяжелого шарика

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

Выполняем GD, перемещаем обновленный x в направлении предыдущего шага на $\beta_k \|x_k - x_{k-1}\|$ единиц.

- Ускорение Нестерова

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k + \beta_k (x_k - x_{k-1})) + \beta_k (x_k - x_{k-1})$$

или

$$y_{k+1} = x_k - \beta_k (x_k - x_{k-1}),$$

$$x_{k+1} = y_{k+1} - \alpha_k \nabla f(y_{k+1}).$$

Перемещаем не обновленный x в направлении предыдущего шага на $\beta_k \|x_k - x_{k-1}\|$ единиц, выполняем GD на сдвинутом x .

Черный ящик

Итерация градиентного спуска:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Черный ящик

Итерация градиентного спуска:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Рассмотрим семейство первого порядка, где

$$\begin{aligned}x^{k+1} &\in x^0 + \text{span} \{ \nabla f(x^0), \nabla f(x^1), \dots, \nabla f(x^k) \} && f - \text{smooth} \\x^{k+1} &\in x^0 + \text{span} \{ g_0, g_1, \dots, g_k \}, \text{ where } g_i \in \partial f(x^i) && f - \text{non-smooth}\end{aligned} \tag{1}$$

Черный ящик

Итерация градиентного спуска:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Рассмотрим семейство первого порядка, где

$$\begin{aligned}x^{k+1} &\in x^0 + \text{span} \{ \nabla f(x^0), \nabla f(x^1), \dots, \nabla f(x^k) \} && f - \text{smooth} \\x^{k+1} &\in x^0 + \text{span} \{ g_0, g_1, \dots, g_k \}, \text{ where } g_i \in \partial f(x^i) && f - \text{non-smooth}\end{aligned} \tag{1}$$

Чтобы построить нижнюю оценку, нам нужно найти функцию f из соответствующего класса, такую что любой метод из семейства 1 будет работать не быстрее нижней оценки.

Гладкий случай

i Theorem

Существует функция f , которая является L -гладкой и выпуклой, так что любой метод 1 для любого $k : 1 \leq k \leq \frac{n-1}{2}$ удовлетворяет:

$$f(x^k) - f^* \geq \frac{3L\|x^0 - x^*\|_2^2}{32(k+1)^2}$$

- Не важно, какой метод первого порядка вы используете, всегда существует функция f , на которой метод будет сходиться не быстрее чем $\mathcal{O}\left(\frac{1}{k^2}\right)$.
- Ключом к доказательству является явное построение специальной функции f .
- Обратите внимание, что эта оценка $\mathcal{O}\left(\frac{1}{k^2}\right)$ не совпадает со скоростью сходимости градиентного спуска $\mathcal{O}\left(\frac{1}{k}\right)$. Два возможных варианта:
 - a. Нижняя оценка не является точной.
 - b. Градиентный метод не является оптимальным для этой задачи.

Гладкий случай

i Theorem

Существует функция f , которая является L -гладкой и выпуклой, так что любой метод 1 для любого $k : 1 \leq k \leq \frac{n-1}{2}$ удовлетворяет:

$$f(x^k) - f^* \geq \frac{3L\|x^0 - x^*\|_2^2}{32(k+1)^2}$$

- Не важно, какой метод первого порядка вы используете, всегда существует функция f , на которой метод будет сходиться не быстрее чем $\mathcal{O}\left(\frac{1}{k^2}\right)$.
- Ключом к доказательству является явное построение специальной функции f .
- Обратите внимание, что эта оценка $\mathcal{O}\left(\frac{1}{k^2}\right)$ не совпадает со скоростью сходимости градиентного спуска $\mathcal{O}\left(\frac{1}{k}\right)$. Два возможных варианта:
 - a. Нижняя оценка не является точной.
 - b. **Градиентный метод не является оптимальным для этой задачи.**

Метод тяжелого шарика для квадратичной задачи

Question

Какая стратегия шага используется для **GD**?

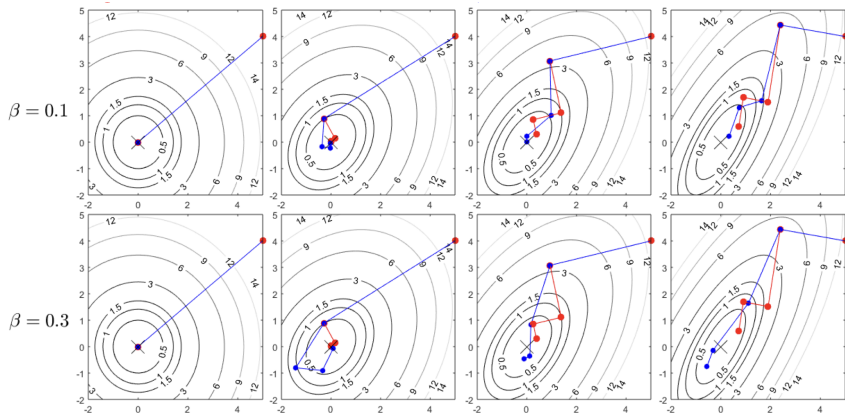


Рис. 1: **GD** vs. **HBM** with fixed β .

Наблюдение: для хорошей f (со сферическими уровнями), GD уже достаточно хорош, и HBM добавляет небольшой эффект. Однако, для плохой f (с эллиптическими уровнями), HBM лучше в некоторых случаях.

Метод тяжелого шарика для квадратичной задачи

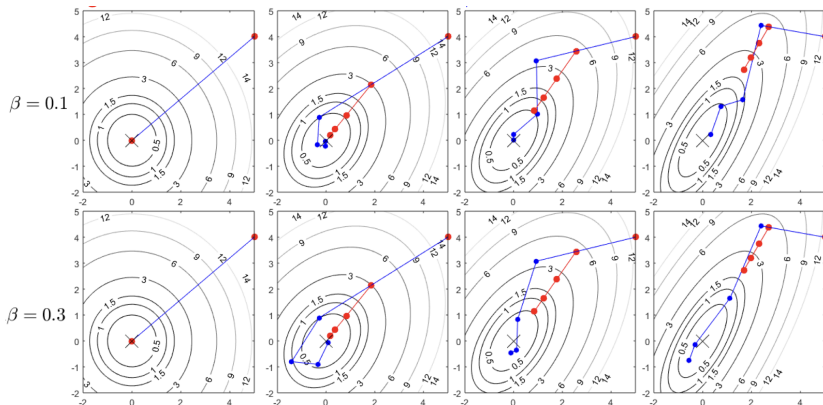


Рис. 2: GD with $\alpha = \frac{1}{L}$ vs. HBM with fixed β .

Наблюдение: то же самое. Если хорошая f (с сферическими уровнями), GD уже достаточно хорош. Если плохая f (с эллиптическими уровнями), HBM лучше в некоторых случаях.

NAG как метод импульса

- Начнем с установки $k = 0, a_0 = 1, x_{-1} = y_0, y_0$ в произвольном параметре, итерации

$$\text{Обновление градиента } x_k = y_k - \alpha_k \nabla f(y_k) \quad (2)$$

$$\text{Вес экстраполяции } a_{k+1} = \frac{1 + \sqrt{1 + 4a_k^2}}{2} \quad (3)$$

$$\text{Экстраполяция } y_{k+1} = x_k + \frac{a_k - 1}{a_{k+1}}(x_k - x_{k-1}) \quad (4)$$

Обратите внимание, что здесь используется фиксированный шаг: $\alpha_k = \frac{1}{L} \forall k$.

Theorem

Если f является L -гладкой и выпуклой, последовательность $\{f(x_k)\}_k$, генерируемая NAG, сходится к оптимальному значению f^* с скоростью $\mathcal{O}(\frac{1}{k^2})$ как

$$f(x_k) - f^* \leq \frac{4L\|x_k - x^*\|^2}{(k+2)^2}$$

NAG для DL ¹

| task | $0_{(SGD)}$ | 0.9N | 0.99N | 0.995N | 0.999N | 0.9M | 0.99M | 0.995M | 0.999M | SGD _C |
|--------|-------------|------|-------------|--------|--------------|------|-------|--------|--------|------------------|
| Curves | 0.48 | 0.16 | 0.096 | 0.091 | 0.074 | 0.15 | 0.10 | 0.10 | 0.10 | 0.16 |
| Mnist | 2.1 | 1.0 | 0.73 | 0.75 | 0.80 | 1.0 | 0.77 | 0.84 | 0.90 | 0.9 |
| Faces | 36.4 | 14.2 | 8.5 | 7.8 | 7.7 | 15.3 | 8.7 | 8.3 | 9.3 | NA |

Рис. 3: Таблица сообщает о квадратичных ошибках на задачах для каждой комбинации β_{max} и типа импульса (NAG=N, HB=M). Когда β_{max} равен 0, выбор между NAG и HB не имеет значения, поэтому ошибки обучения представлены в одном столбце. Для каждого выбора β_{max} используется наиболее эффективный шаг обучения. Столбец SGD_C содержит результаты Chapelle & Erhan (2011), которые использовали 1,7 млн. шагов SGD и сети tanh.

¹Ссылка

Сходимость метода тяжелого шарика ²

i Theorem

Предположим, что f является гладкой и выпуклой и что

$$\beta \in [0, 1), \quad \alpha \in \left(0, \frac{2(1-\beta)}{L}\right).$$

Тогда, последовательность $\{x_k\}$, генерируемая итерацией тяжелого шарика, удовлетворяет

$$f(\bar{x}_T) - f^* \leq \begin{cases} \frac{\|x_0 - x^*\|^2}{2(T+1)} \left(\frac{L\beta}{1-\beta} + \frac{1-\beta}{\alpha} \right), & \text{if } \alpha \in (0, \frac{1-\beta}{L}], \\ \frac{\|x_0 - x^*\|^2}{2(T+1)(2(1-\beta) - \alpha L)} \left(L\beta + \frac{(1-\beta)^2}{\alpha} \right), & \text{if } \alpha \in [\frac{1-\beta}{L}, \frac{2(1-\beta)}{L}), \end{cases}$$

где \bar{x}_T является средним по Чезаро итераций, т.е.

$$\bar{x}_T = \frac{1}{T+1} \sum_{k=0}^T x_k.$$

²Сходимость метода тяжелого шарика для выпуклой оптимизации, Euhanna Ghadimi et.al.

Сходимость метода тяжелого шарика³

i Theorem

Предположим, что f является гладкой и сильно выпуклой и что

$$\alpha \in (0, \frac{2}{L}), \quad 0 \leq \beta < \frac{1}{2} \left(\frac{\mu\alpha}{2} + \sqrt{\frac{\mu^2\alpha^2}{4} + 4(1 - \frac{\alpha L}{2})} \right).$$

Тогда, последовательность $\{x_k\}$, генерируемая итерацией тяжелого шарика, сходится линейно к уникальному оптимальному значению x^* . В частности,

$$f(x_k) - f^* \leq q^k(f(x_0) - f^*),$$

где $q \in [0, 1)$.

³Сходимость метода тяжелого шарика для выпуклой оптимизации, Euhanna Ghadimi et.al.

Сходимость NAG

i Theorem

Предположим, что $f : \mathbb{R}^n \rightarrow \mathbb{R}$ является выпуклой и L -гладкой. Алгоритм Nesterov Accelerated Gradient Descent (NAG) предназначен для решения задачи минимизации, начиная с начальной точки $x_0 = y_0 \in \mathbb{R}^n$ и $\lambda_0 = 0$. Алгоритм выполняет следующие шаги:

| | |
|------------------------------|---|
| Экстраполяция: | $y_{k+1} = x_k + \beta(x_k - x_{k-1})$ |
| Обновление градиента: | $x_{k+1} = y_{k+1} - \frac{1}{L} \nabla f(y_{k+1})$ |
| Вес экстраполяции: | $\lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2}$ |
| Вес экстраполяции: | $\beta_k = \frac{1 - \lambda_k}{\lambda_{k+1}}$ |

Последовательности $\{f(x_k)\}_{k \in \mathbb{N}}$, генерируемые алгоритмом, сходятся к оптимальному значению f^* с скоростью $\mathcal{O}\left(\frac{1}{k^2}\right)$, в частности:

$$f(x_k) - f^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}$$

i Theorem

Предположим, что $f : \mathbb{R}^n \rightarrow \mathbb{R}$ является μ -сильно выпуклой и L -гладкой. Алгоритм Нестерова (NAG) предназначен для решения задачи минимизации, начиная с начальной точки $x_0 = y_0 \in \mathbb{R}^n$ и $\lambda_0 = 0$. Алгоритм выполняет следующие шаги:


Экстраполяция: $y_{k+1} = x_k + \beta(x_k - x_{k-1})$

Обновление градиента: $x_{k+1} = y_{k+1} - \frac{1}{L} \nabla f(y_{k+1})$


Вес экстраполяции: $\gamma_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$

Последовательности $\{f(x_k)\}_{k \in \mathbb{N}}$, генерируемые алгоритмом, сходятся к оптимальному значению f^* линейно:



$$f(x_k) - f^* \leq \frac{\mu + L}{2} \|x_0 - x^*\|_2^2 \exp\left(-\frac{k}{\sqrt{\kappa}}\right)$$

Давайте напишем код!  Colab

Логистическая регрессия

Давайте напишем код!  Colab

Ссылки и примеры Python

- Изображения для HBM взяты из презентации. Посетите сайт для большего количества туториалов.
- Почему импульс действительно работает. Ссылка.
- Запустите код в  Colab. Код взят из .
- Важность инициализации и импульса в глубоком обучении. Ссылка.

NAG для квадратичной задачи

Рассмотрим следующую квадратичную задачу оптимизации:

$$\min_{x \in \mathbb{R}^d} q(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x, \text{ where } A \in \mathbb{S}_{++}^d.$$

Каждая симметричная матрица A имеет разложение на собственные значения

$$A = Q \text{diag}(\lambda_1, \dots, \lambda_n) Q^\top = Q \Lambda Q^\top, \quad Q = [q_1, \dots, q_n].$$

и, как по соглашению, мы будем считать, что λ_i 's отсортированы, от наименьшего λ_1 до наибольшего λ_n .
Очевидно, что λ_i соответствует **кривизне** вдоль соответствующих направлений собственных векторов.

Мы можем перепараметризовать $q(x)$ с помощью матричного преобразования Q и оптимизировать $y = Qx$ с помощью целевой функции

$$p(y) \equiv q(x) = q(Q^\top y) = y^\top Q(Q^\top \Lambda Q) Q^\top y / 2 - b^\top Q^\top y = y^\top \Lambda y / 2 - c^\top y,$$

где $c = Qb$.

Мы можем еще раз переписать p как

$$p(y) = \sum_{i=1}^n [p]_i([y]_i),$$

где $[p]_i(t) = \lambda_i t^2 / 2 - [c]_i t$.

NAG для квадратичной задачи

💡 Теорема 2.1 из [1].

Пусть $p(y) = \sum_{i=1}^n [p]_i([y]_i)$ такой, что $[p]_i(t) = \lambda_i t^2/2 - [c]_i t$. Пусть α будет произвольным и фиксированным. Обозначим через $\text{HBM}_x(\beta, p, y, v)$ и $\text{HBM}_v(\beta, p, y, v)$ вектор параметров и вектор скорости соответственно, полученные применением одного шага HBM (т.е. уравнения 1 и затем уравнения 2) к функции p в точке y , со скоростью v , коэффициентом импульса β и шагом обучения α . Определим NAG_x и NAG_v аналогично. Тогда для $z \in \{x, v\}$:

$$\text{HBM}_z(\beta, p, y, v) = \begin{bmatrix} \text{HBM}_z(\beta, [p]_1, [y]_1, [v]_1) \\ \vdots \\ \text{HBM}_z(\beta, [p]_n, [y]_n, [v]_n) \end{bmatrix}$$

$$\text{NAG}_z(\beta, p, y, v) = \begin{bmatrix} \text{HBM}_z(\beta(1 - \alpha\lambda_1), [p]_1, [y]_1, [v]_1) \\ \vdots \\ \text{HBM}_z(\beta(1 - \alpha\lambda_n), [p]_n, [y]_n, [v]_n) \end{bmatrix}$$

NAG для квадратичной задачи. Доказательство (1/2)

Доказательство:

Легко показать, что если

$$x_{i+1} = \text{HBM}_x(\beta_i, [q]_i, [x]_i, [v]_i)$$

$$v_{i+1} = \text{HBM}_v(\beta_i, [q]_i, [x]_i, [v]_i)$$

то для $y_i = Qx_i, w_i = Qv_i$

$$y_{i+1} = \text{HBM}_x(\beta_i, [p]_i, [y]_i, [w]_i)$$

$$w_{i+1} = \text{HBM}_v(\beta_i, [p]_i, [y]_i, [w]_i)$$

. Тогда, рассмотрим один шаг HBM_v :

$$\begin{aligned}\text{HBM}_v(\beta, p, y, v) &= \beta v - \alpha \nabla p(y) \\ &= (\beta[v]_1 - \alpha \nabla_{[y]_1} p(y), \dots, \beta[v]_n - \alpha \nabla_{[y]_n} p(y)) \\ &= (\beta[v]_1 - \alpha \nabla[p]_1([y]_1), \dots, \beta[v]_n - \alpha \nabla[p]_n([y]_n)) \\ &= (\text{HBM}_v(\beta_1, [p]_1, [y]_1, [v]_1), \dots, \text{HBM}_v(\beta_i, [p]_i, [y]_i, [v]_i))\end{aligned}$$

Это показывает, что один шаг HBM_v на p точно эквивалентен n одновременным применениям HBM_v к одномерным квадратичным $[p]_i$, все с одним и тем же β и α . Аналогично, для HBM_x .

NAG для квадратичной задачи. Доказательство (2/2)

Далее мы покажем, что NAG, примененный к одномерной квадратичной задаче с коэффициентом импульса β , эквивалентен HBM, примененному к той же квадратичной задаче и с тем же шагом обучения, но с коэффициентом импульса $\beta(1 - \alpha\lambda)$. Мы покажем это, раскрыв $\text{NAG}_v(\beta, [p]_i, y, v)$ (где y и v являются скалярами):

$$\begin{aligned}\text{NAG}_v(\beta, [p]_i, y, v) &= \beta v - \alpha \nabla [p]_i(y + \beta v) \\ &= \beta v - \alpha(\lambda_i(y + \beta v) - c_i) \\ &= \beta v - \alpha\lambda_i\beta v - \alpha(\lambda_i y - c_i) \\ &= \beta(1 - \alpha\lambda_i)v - \alpha \nabla [p]_i(y) \\ &= \text{HBM}_v(\beta(1 - \alpha\lambda_i), [p]_i, y, v).\end{aligned}$$

ч.т.д.

Наблюдения:

- HBM и NAG становятся **эквивалентными** когда α мал (когда $\alpha\lambda \ll 1$ для каждого собственного значения λ матрицы A), поэтому NAG и HBM отличаются только когда α достаточно велико.
- Когда α относительно велико, NAG использует меньший эффективный импульс для направлений с высокой кривизной, что **предотвращает колебания** (или расхождение) и, таким образом, позволяет использовать большее β , что допускает СМ при заданном α .