

Коллоквиум по МОМО

immediate

2023-2024 учебный год

Определения

0.1 Положительно определённая матрица

Матрица $A \in \mathbb{S}$ называется положительно (отрицательно) определённой, если для $\forall x \neq 0 : x^T A x > (<)0$.

Обозначение: $A \prec 0$ ($A \succ 0$).

Аналогично определяется полуопределённость, только там неравенства нестрогие.

0.2 Евклидова норма вектора

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$$

Данная норма соответствует расстоянию в реальном мире. Иначе называется 2-норма (см. р-норма вектора)

0.3 Неравенство треугольника для нормы

Норма должна удовлетворять следующим свойствам:

1. $\|\alpha x\| = |\alpha| \|x\|, \alpha \in \mathbb{R}$
2. $\|x\| = 0 \Rightarrow x = 0$
3. $\|x + y\| \leq \|x\| + \|y\|$ – неравенство треугольника

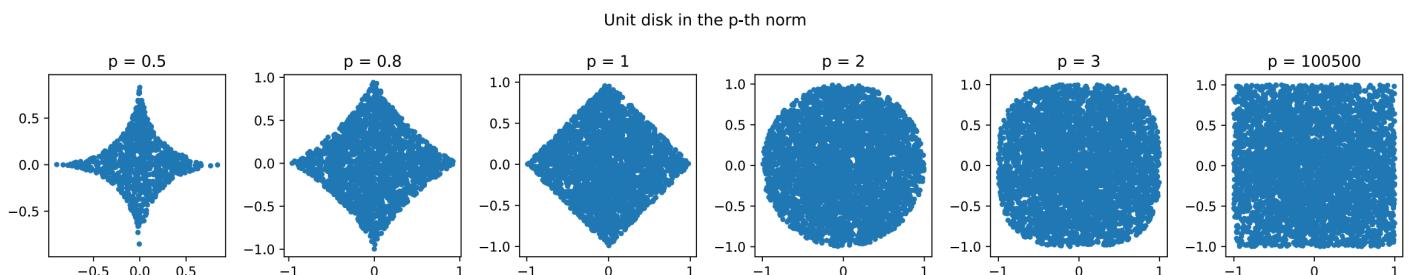
0.4 p -норма вектора

$$\|x\|_p = \left(\sum_{i=0}^n |x_i|^p \right)^{\frac{1}{p}}$$

Важные частные случаи:

- Норма Чебышева: $\|x\|_\infty = \max_i |x_i|$
- Манхэттенское расстояние или $L1$ норма: $\|x\|_1 = \sum_{i=0}^n |x_i|$

0.5 Как выглядит единичный шар в p -норме на плоскости для $p = 1, 2, \infty$?



0.6 Норма Фробениуса для матрицы

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}}$$

(смысл: корень суммы квадратов всех элементов матрицы A)

0.7 Спектральная норма матрицы

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_1(A) = \sqrt{\lambda_{\max}(A^\top A)}$$

Где $\sigma_1(A)$ – старшее сингулярное значение A , $\lambda_{\max}(A^\top A)$ – наибольшее собственное значение $A^\top A$.

0.8 Скалярное произведение двух векторов

Пусть $x, y \in \mathbb{R}^n$, тогда их скалярное произведение это

$$\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i = y^T x = \langle y, x \rangle$$

0.9 Скалярное произведение двух матриц, согласованное с нормой Фробениуса

Пусть $X, Y \in \mathbb{R}^{m \times n}$, тогда их скалярное произведение это

$$\langle X, Y \rangle = \text{tr}(X^T Y) = \sum_{i=1}^m \sum_{j=1}^n X_{ij} Y_{ij} = \text{tr}(Y^T X) = \langle Y, X \rangle$$

Связь с нормой Фробениуса: $\langle X, X \rangle = \|X\|_F^2$

0.10 Спектр матрицы

Скаляр λ является собственным значением для матрицы A , если существует ненулевой вектор q , такой что $Aq = \lambda q$. В таком случае q называют собственным вектором.

Спектр матрицы – совокупность её собственных значений.

0.11 Связь спектра матрицы и её определенности

Матрица положительно (неотрицательно) определена \iff её спектр (все её собственные значения) положительны (неотрицательны).

0.12 Спектральное разложение матрицы

Спектральное разложение матрицы, или разложение матрицы на основе собственных векторов, — это представление квадратной матрицы A в виде произведения трёх матриц $A = V \Lambda V^{-1}$, где V — матрица, столбцы которой

являются собственными векторами матрицы A , Λ — диагональная матрица с соответствующими собственными значениями на главной диагонали. В таком виде могут быть представлены только матрицы, обладающие полным набором собственных векторов.

Тогда $A^n = V\Lambda^n V^{-1}$.

(полный набор собственных векторов — набор из n линейно независимых собственных векторов, где n — это порядок матрицы)

0.13 Сингулярное разложение матрицы

$A \in \mathbb{R}^{m \times n}$, $\text{rank } A = r$.

$$A = U\Sigma V^T, \quad (A.12)$$

$U \in \mathbb{R}^{m \times r}$, $U^T U = I$, $V \in \mathbb{R}^{n \times r}$, $V^T V = I$, Σ is a diagonal matrix with

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r),$$

such that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0,$$

Столбцы U , V — левые и правые собственные векторы A , σ_i — сингулярные значений.

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T,$$

0.14 Связь определителя и собственных чисел для квадратной матрицы

если у матрицы A собственные значения $\lambda_1, \lambda_2, \dots, \lambda_n$, то ее определитель равен:

$$\det(A) = \lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_n$$

0.15 Связь следа и собственных чисел для квадратной матрицы

если у матрицы A собственные значения $\lambda_1, \lambda_2, \dots, \lambda_n$, то ее след равен:

$$\text{tr}(A) = \lambda_1 + \lambda_2 + \dots + \lambda_n$$

0.16 Линейная сходимость последовательности

Пусть есть последовательность $\{\|x_k - x^*\|_2\}$ в \mathbb{R} , сходящаяся к 0.

Линейная сходимость при $q \in (0, 1)$ (скорость сходимости) и $C \in (0, \infty)$ (константа сходимости) определяется одним из двух способов:

$$\|x_{k+1} - x^*\| \leq Cq^k \text{ или } \|x_{k+1} - x^*\| \leq q\|x_k - x^*\|$$

Чем меньше q , тем быстрее сходится последовательность.

По-другому, говорят что последовательность x_k сходится к числу L . Мы говорим, что эта последовательность линейно сходится к L , если \exists число $\mu \in (0,1)$, такое, что

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - L|}{|x_k - L|} = \mu$$

и μ называется скоростью сходимости.

0.17 Сублинейная сходимость последовательности

Если последовательность r_k сходится к нулю, но не обладает линейной сходимостью, то говорят, что она сходится сублинейно. Иногда мы можем рассматривать следующий класс сублинейной сходимости:

$$|x_{k+1} - x^*|_2 \leq C k^q,$$

где $q < 0$ и $0 < C < \infty$.

0.18 Сверхлинейная сходимость последовательности

Сверхлинейная сходимость при $q > 1$, $C > 0$ определяется следующим образом:

$$|x_{k+1} - x^*| \leq C |x_k - x^*|^q$$

0.19 Квадратичная сходимость последовательности

Квадратичная сходимость является частным случаем сверхлинейной сходимости, когда $q = 2$. Она определяется следующим образом:

$$|x_{k+1} - x^*| \leq C |x_k - x^*|^2$$

Или по-другому:

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - L|}{|x_k - L|^2} = \mu$$

где $\mu > 0$.

0.20 Тест корней для определения скорости сходимости последовательности

Пусть $(r_k)_{k=m}^\infty$ - последовательность неотрицательных чисел, сходящаяся к нулю, и пусть $\alpha := \limsup_{k \rightarrow \infty} r_k^{1/k}$. (Заметим, что $\alpha \geq 0$.)

- (a) Если $0 \leq \alpha < 1$, то $(r_k)_{k=m}^\infty$ сходится линейно с константой α .
- (b) В частности, если $\alpha = 0$, то $(r_k)_{k=m}^\infty$ сходится сверхлинейно.
- (c) Если $\alpha = 1$, то $(r_k)_{k=m}^\infty$ сходится сублинейно.
- (d) Случай $\alpha > 1$ невозможен.

0.21 Тест отношений для определения скорости сходимости последовательности

Пусть $r_{k=m}^\infty$ - последовательность строго положительных чисел, сходящаяся к нулю. Пусть

$$q = \lim_{k \rightarrow \infty} \frac{r_{k+1}}{r_k}$$

1. Если существует q и $0 \leq q < 1$, то $r_{k=m}^\infty$ имеет линейную сходимость с константой q .
2. В частности, если $q = 0$, то $r_{k=m}^\infty$ имеет сверхлинейную сходимость.
3. Если q не существует, но $q = \limsup_{k \rightarrow \infty} \frac{r_{k+1}}{r_k} < 1$, то $r_{k=m}^\infty$ имеет линейную сходимость с константой, не превышающей q .
4. Если $\liminf_{k \rightarrow \infty} \frac{r_{k+1}}{r_k} = 1$, то $r_{k=m}^\infty$ имеет сублинейную сходимость.
5. Случай $\liminf_{k \rightarrow \infty} \frac{r_{k+1}}{r_k} > 1$ невозможен.

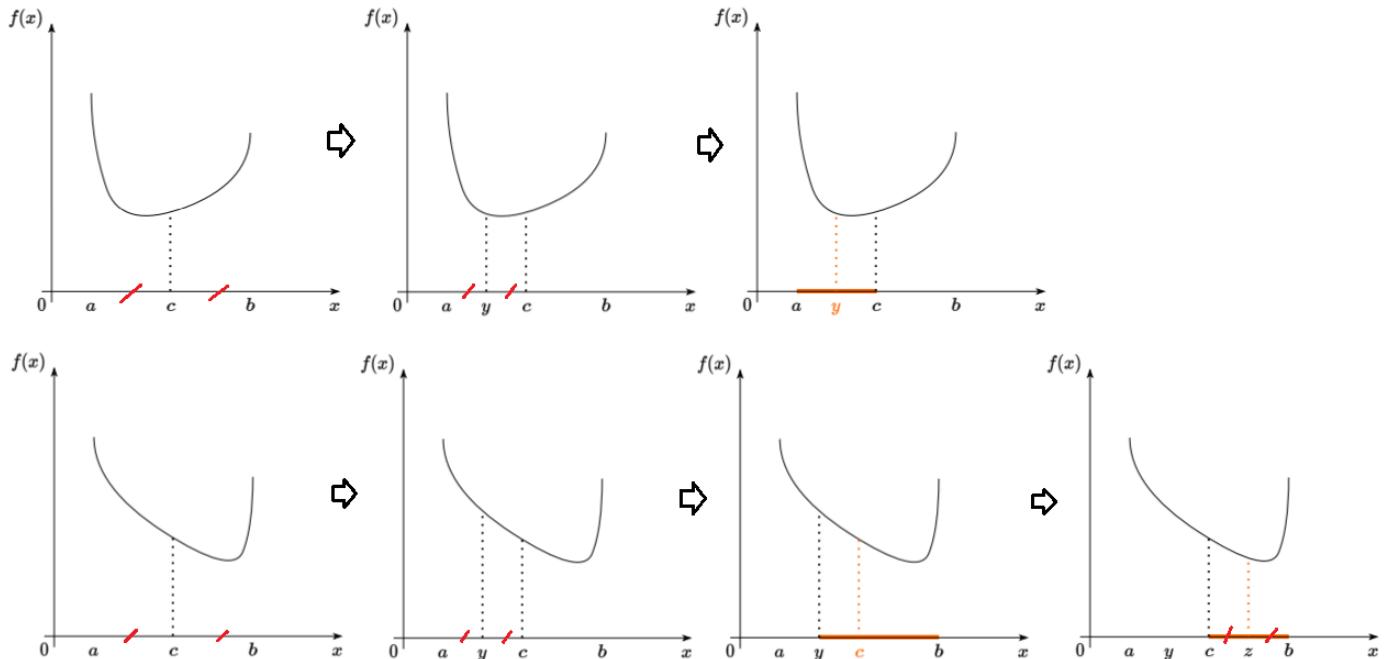
0.22 Унимодальная функция

Функция $f(x)$ называется унимодальной на $[a, b]$, если существует $x^* \in [a, b]$, такое, что

1. $f(x_1) > f(x_2)$ для всех $a \leq x_1 < x_2 < x^*$
2. $f(x_1) < f(x_2)$ для всех $x^* < x_1 < x_2 \leq b$

0.23 Метод дихотомии

Наша цель - решить следующую задачу: $\min_{x \in [a, b]} f(x)$ Мы делим отрезок на две равные части и выбираем ту, которая содержит решение задачи, используя значения функции, опираясь на ключевое свойство, описанное выше. Наша цель после одной итерации метода - уменьшить область поиска решения в два раза (в среднем). Метод описан на рисунках ниже.



Длина отрезка на $(k + 1)$ -ой итерации:

$$\Delta_{k+1} = b_{k+1} - a_{k+1} = \frac{1}{2^k}(b - a)$$

Для унимодальных функций:

$$|x_{k+1} - x^*| \leq \frac{\Delta_{k+1}}{2} \leq \frac{1}{2^{k+1}}(b - a) \leq (0.5)^{k+1} \cdot (b - a)$$

Заметим, что на каждой итерации мы обращаемся к оракулу не более чем два раза, поэтому число вычислений функции равно $N = 2 \cdot k$, что подразумевает:

$$|x_{k+1} - x^*| \leq (0.5)^{\frac{N}{2}+1} \cdot (b - a) \leq (0.707)^N \frac{b - a}{2}$$

0.24 Метод золотого сечения

Общая идея: хотим поделить отрезок на 3 части так, чтобы потом когда одна из частей отпадет на следующей итерации одно из нужных значений функций будет уже известно.

```
def golden_search(f, a, b, epsilon):
    tau = (sqrt(5) + 1) / 2
    y = a + (b - a) / tau**2
    z = a + (b - a) / tau
    while b - a > epsilon:
        if f(y) <= f(z):
            b = z
            z = y
            y = a + (b - a) / tau**2
        else:
            a = y
            y = z
            z = a + (b - a) / tau
    return (a + b) / 2
```

0.25 Метод параболической интерполяции

Идея метода: берем 3 точки, по этим 3 точкам однозначно строим параболу, находим ее минимум, и из этих 4 точек оставляем 3 так, чтобы между первой и третьей находился минимум.

```
def parabola_search(f, x1, x2, x3, epsilon):
    f1, f2, f3 = f(x1), f(x2), f(x3)
    while x3 - x1 > epsilon:
        u = x2 - ((x2 - x1)**2 * (f2 - f3) - (x2 - x3)**2 * (f2 - f1)) / (2 * ((x2 - x1) * (f2 - f3) - (x2 - x3) * (f2 - f1)))
        fu = f(u)
```

```

if x2 <= u:
    if f2 <= fu:
        x1, x2, x3 = x1, x2, u
        f1, f2, f3 = f1, f2, fu
    else:
        x1, x2, x3 = x2, u, x3
        f1, f2, f3 = f2, fu, f3
else:
    if fu <= f2:
        x1, x2, x3 = x1, u, x2
        f1, f2, f3 = f1, fu, f2
    else:
        x1, x2, x3 = u, x2, x3
        f1, f2, f3 = fu, f2, f3
return (x1 + x3) / 2

```

0.26 Условие достаточного убывания для неточного линейного поиска

Неточный линейный поиск:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) \alpha = \operatorname{argmin} f(x_{k+1})$$

Хотим приближенно найти α . Сведем задачу к поиску минимума следующей функции:

$$\phi(\alpha) = f(x_k - \alpha \nabla f(x_k)), \alpha \geq 0$$

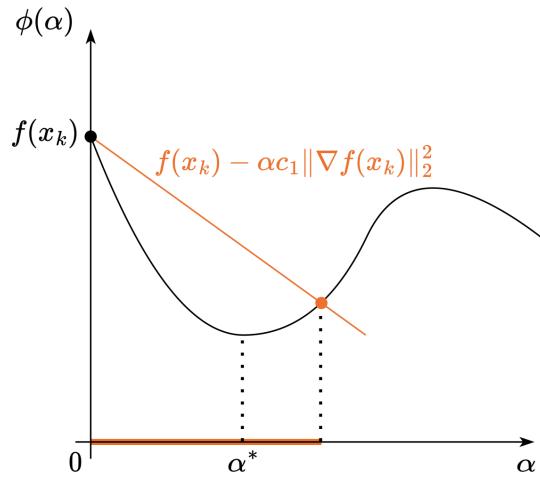
Приблизим ее через первые 2 члена ряда Тейлора:

$$\phi(\alpha) \approx f(x_k) - \alpha \nabla f(x_k)^\top \nabla f(x_k)$$

Тогда условием достаточного убывания (Armijo condition) является:

$$f(x_k - \alpha \nabla f(x_k)) \leq f(x_k) - c_1 \cdot \alpha \nabla f(x_k)^\top \nabla f(x_k), c_1 \in (0, 1)$$

Иллюстрация для понимания:



0.27 Условия Гольдштейна для неточного линейного поиска

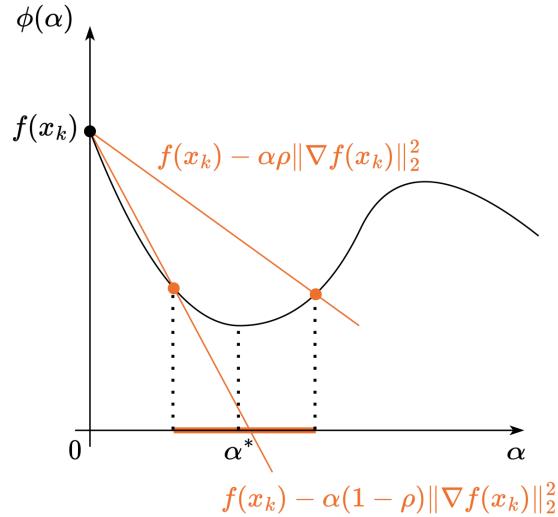
Определим ϕ_1 и ϕ_2 следующим образом ($c_1 > c_2$)

$$\phi_1(\alpha) = f(x_k) - c_1 \alpha \|\nabla f(x_k)\|^2$$

$$\phi_2(\alpha) = f(x_k) - c_2 \alpha \|\nabla f(x_k)\|^2$$

Тогда условие Гольдштейна заключается в том, что $\phi_1(\alpha) \leq \phi(\alpha) \leq \phi_2(\alpha)$.

Иллюстрация для понимания:

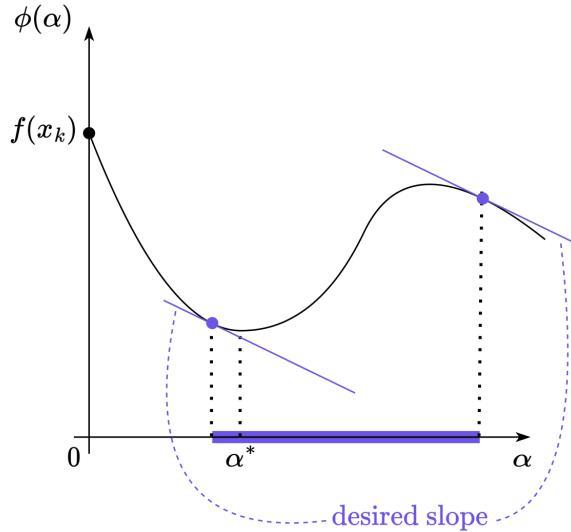


0.28 Условие ограничения на кривизну для неточного линейного поиска

$$-\nabla f(x_k - \alpha \nabla f(x_k))^\top \nabla f(x_k) \geq c_2 \nabla f(x_k)^\top (-\nabla f(x_k)),$$

где $c_2 \in (c_1, 1)$, и c_1 взято из условия достаточного убывания.

Иллюстрация для понимания:



0.29 Градиент функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$

$\nabla f(x)$, вектор частных производных функции f .

0.30 Гессиан функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$

$$f''(x) = \nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$

0.31 Якобиан функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$J_f = f'(x) = \frac{df}{dx^T} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

0.32 Формула для аппроксимации Тейлора первого порядка $f_{x_0}^I(x)$ функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ в точке x_0

Для дифференцируемой f :

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T(x - x_0)$$

0.33 Формула для аппроксимации Тейлора второго порядка $f_{x_0}^{II}(x)$ функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ в точке x_0

Для дважды дифференцируемой f :

$$f_{x_0}^{II}(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0)$$

0.34 Связь дифференциала функции df и градиента ∇f для функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$

$$df(x) = \langle \nabla f(x), dx \rangle$$

0.35 Связь второго дифференциала функции $d^2 f$ и гессиана $\nabla^2 f$ для функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$

$$d(df) = d^2 f(x) = \langle \nabla^2 f(x) dx_1, dx \rangle = \langle H_f(x) dx_1, dx \rangle$$

0.36 Формула для приближенного вычисления производной функции $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ по k -ой координате с помощью метода конечных разностей

$$\frac{\partial f}{\partial x_k}(x) \approx \frac{f(x + \varepsilon e_k) - f(x)}{\varepsilon}, \quad e_k = (0, \dots, \underset{k}{1}, \dots, 0)$$

Время работы: $2dT$, где вызов $f(x)$ занимает $T, x \in \mathbb{R}^d$

0.37 Пусть $f = f(x_1(t), \dots, x_n(t))$. Формула для вычисления $\frac{\partial f}{\partial t}$ через $\frac{\partial x_i}{\partial t}$ (Forward chain rule)

$$\frac{\partial f}{\partial t} = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{\partial x_i}{\partial t}$$

0.38 Пусть L - функция, возвращающая скаляр, а v_k - функция, возвращающая вектор $x \in \mathbb{R}^t$. Формула для вычисления $\frac{\partial L}{\partial v_k}$ через $\frac{\partial L}{\partial x_i}$ (Backward chain rule)

$$\frac{\partial L}{\partial v_k} = \sum_{i=1}^t \frac{\partial L}{\partial x_i} \frac{\partial x_i}{\partial v_k}$$

0.39 Идея Хатчинсона для оценки следа матрицы с помощью matvec операций

$X \in \mathbb{R}^{d \times d}$, $v \in \mathbb{R}^d$ - случайный вектор: $\mathbb{E}[vv^T] = I$.

$$\text{Tr}(X) = \mathbb{E}[v^T X v] = \frac{1}{V} \sum_{i=1}^V v_i^T X v_i.$$

0.40 Аффинное множество. Аффинная комбинация. Аффинная оболочка

Множество A называется аффинным если для любых x_1, x_2 из A прямая, проходящая через x_1, x_2 , тоже лежит в A . То есть:

$$\forall \theta \in \mathbb{R}, \forall x_1, x_2 \in A : \theta x_1 + (1 - \theta) x_2 \in A$$

Пример аффинного множества: \mathbb{R}^n

Пусть $x_1, x_2, \dots, x_k \in S$. Тогда точка $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ называется аффинной комбинацией, если

$$\forall i \in \{1, \dots, k\} : \theta_i \in \mathbb{R}, \quad \sum_{i=1}^k \theta_i = 1$$

Аффинная оболочка – множество всех возможных аффинных комбинаций элементов множества.

$$\text{aff}(S) = \left\{ \sum_{i=1}^k \theta_i x_i \mid k > 0, x_i \in S, \theta_i \in \mathbb{R}, \sum_{i=1}^k \theta_i = 1 \right\}$$

0.41 Выпуклое множество. Выпуклая комбинация. Выпуклая оболочка

Множество S называется выпуклым если для любых x_1, x_2 из S отрезок между x_1, x_2 тоже лежит в S . То есть:

$$\forall \theta \in [0, 1], \forall x_1, x_2 \in S : \theta x_1 + (1 - \theta) x_2 \in S$$

Пусть $x_1, x_2, \dots, x_k \in S$. Тогда точка $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ называется выпуклой комбинацией, если

$$\forall i \in \{1, \dots, k\} : \theta_i \geq 0, \quad \sum_{i=1}^k \theta_i = 1$$

Выпуклая оболочка – множество всех возможных выпуклых комбинаций элементов множества.

$$\text{conv}(S) = \left\{ \sum_{i=1}^k \theta_i x_i \mid k > 0, x_i \in S, \theta_i \geq 0, \sum_{i=1}^k \theta_i = 1 \right\}$$

0.42 Конус. Выпуклый конус. Коническая комбинация. Коническая оболочка

Множество S называется конусом если для любого x из S луч, проходящий из 0 через x , тоже лежит в S . То есть:

$$\forall \theta \geq 0, \forall x \in S : \theta x \in S$$

Множество S называется выпуклым конусом если для любых $x_1, x_2 \in S$ их коническая комбинация тоже лежит в S . То есть:

$$\forall x_1, x_2 \in S, \theta_1, \theta_2 \geq 0 : \theta_1 x_1 + \theta_2 x_2 \in S$$

Пусть $x_1, x_2, \dots, x_k \in S$. Тогда точка $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ называется конической комбинацией, если

$$\forall i \in \{1, \dots, k\} : \theta_i \geq 0$$

Коническая оболочка – множество всех возможных конических комбинаций элементов множества.

$$\text{coni}(S) = \left\{ \sum_{i=1}^k \theta_i x_i \mid k > 0, x_i \in S, \theta_i \geq 0 \right\}$$

0.43 Внутренность множества

Внутренность множества – объединение всех открытых подмножеств данного множества. Точки внутренности называются внутренними точками.

0.44 Относительная внутренность множества

Относительная внутренность множества – внутренность множества в его аффинной оболочке. Может быть полезной при работе с множествами меньшей размерности чем пространство, в котором они находятся.

$$\text{relint}(S) := \{x \in S \mid \exists \varepsilon > 0, N_\varepsilon(x) \cap \text{aff}(S) \subseteq S\}$$

$N_\varepsilon(x)$ – шар радиуса ε с центром в x , $\text{aff}(S)$ – аффинная оболочка S

Пример: отрезок на плоскости имеет пустую внутренность, но его относительная внутренность – тот же отрезок без концов.

0.45 Сумма Минковского

Сумма Минковского – евклидово пространство, формирующееся сложением каждого вектора из S_1 с каждым вектором из S_2 :

$$S_1 + S_2 = \{s_1 + s_2 \mid s_1 \in S_1, s_2 \in S_2\}$$

0.46 Любые 2 операции с множествами, сохраняющие выпуклость

1. Линейная комбинация:

$$S = \{s \mid s = c_1x + c_2y, x \in S_x, y \in S_y, c_1, c_2 \in \mathbb{R}\}$$

2. Пересечение любого числа выпуклых множеств

3. Образ множества в аффинном преобразовании:

$$S \subseteq \mathbb{R}^n \text{ convex} \rightarrow f(S) = \{f(x) \mid x \in S\} \text{ convex} \quad (f(x) = Ax + b)$$

0.47 Выпуклая функция

Функция $f(x)$, определённая на выпуклом множестве $S \subseteq \mathbb{R}^n$ называется выпуклой на S если:

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1]$$

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

0.48 Строго выпуклая функция

Функция $f(x)$, определённая на выпуклом множестве $S \subseteq \mathbb{R}^n$ называется строго выпуклой на S если:

$$\forall x_1, x_2 \in S : x_1 \neq x_2, \quad \forall \lambda \in (0, 1)$$

$$f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2)$$

0.49 Надграфик функции $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$

Для функции, определённой на $S \subseteq \mathbb{R}^n$, множество:

$$\text{epi } f = \{[x, \mu] \in S \times \mathbb{R} : f(x) \leq \mu\}$$

называется надграфиком функции $f(x)$

0.50 Множество подуровней функции $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$

Для функции, определённой на $S \subseteq \mathbb{R}^n$, множество:

$$\mathcal{L}_\beta = \{x \in S : f(x) \leq \beta\}$$

называется множеством подуровней или множеством Лебега функции $f(x)$

Если множество подуровней выпукло \iff функция выпукла.

0.51 Дифференциальный критерий выпуклости первого порядка.

Дифференцируемая функция определенная на выпуклом множестве $S \subseteq \mathbb{R}^n$ выпукла тогда и только тогда когда $\forall x, y \in S$:

$$f(y) \geq f(x) + \nabla f^T(x)(y - x)$$

0.52 Дифференциальный критерий выпуклости второго порядка.

Дважды дифференцируемая функция определенная на выпуклом множестве $S \subseteq \mathbb{R}^n$ выпукла тогда и только тогда когда для любой внутренней точки x $\forall x \in \text{int}(S) \neq \emptyset$:

$$\nabla^2 f(x) \succeq 0$$

0.53 Связь выпуклости функции и её надграфика.

Функция выпукла тогда и только тогда, когда её надграфик - выпуклое множество.

0.54 μ -сильно выпуклая функция.

Функция определенная на выпуклом множестве $S \subseteq \mathbb{R}^n$ называется сильно выпуклой если $\forall x_1, x_2 \in S$, $0 \leq \lambda \leq 1$ и $\mu > 0$:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) - \frac{\mu}{2}\lambda(1 - \lambda)\|x_1 - x_2\|^2$$

0.55 Дифференциальный критерий сильной выпуклости первого порядка.

Дифференцируемая функция определенная на выпуклом множестве $S \subseteq \mathbb{R}^n$ является сильно выпуклой тогда и только тогда, когда $\forall x, y \in S$:

$$f(y) \geq f(x) + \nabla f^T(x)(y - x) + \frac{\mu}{2}\|y - x\|^2$$

0.56 Дифференциальный критерий сильной выпуклости второго порядка.

Дважды дифференцируемая функция определенная на выпуклом множестве $S \subseteq \mathbb{R}^n$ является сильно выпуклой тогда и только тогда, когда

$$\nabla^2 f(x) \succeq \mu I$$

0.57 Любые 2 операции с функциями, сохраняющие выпуклость.

- Сумма выпуклых функций с не отрицательными коэффициентами является выпуклой функцией.
- Композиция выпуклой функции с афинной выпукла: $g(x) = f(Ax + b)$
- Поточечный максимум любого числа выпуклых функций есть выпуклая функция.

0.58 Теорема Тейлора.

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ - непрерывная, дифференцируемая функция и $p \in \mathbb{R}^n$, тогда теорема Тейлора гласит:

$$f(x + p) = f(x) + \nabla f(x + tp)^T p$$

Для некоторого $t \in (0, 1)$

Более того, если f - дважды дифференцируема, то:

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

Для некоторого $t \in (0, 1)$

0.59 Необходимые условия локального экстремума.

Если x^* - локальный экстремум и f непрерывная дифференцируема в открытой окрестности x^* , то:

$$\nabla f(x^*) = 0$$

0.60 Достаточные условия локального экстремума.

Если $\nabla^2 f$ непрерывна в открытой окрестности x^* и

$$\nabla f(x^*) = 0$$

$$\nabla^2 f(x^*) \succ 0$$

То x^* - локальный минимум $f(x)$.

Для локального максимума аналогично, только

$$0 \succ \nabla^2 f(x^*)$$

0.61 Общая задача математического программирования. Функция Лагранжа

Общая задача М. п. состоит в нахождении оптимального (максимального или минимального) значения целевой функции, причем значения переменных должны принадлежать некоторой области допустимых значений.

Пример задачи с условиями на равенство:

$$f(x) \rightarrow \min_{x \in \mathbb{R}^n}$$

$$s.t. h(x) = 0$$

Функция Лагранжа будет выглядеть:

$$L(x, \nu) = f(x) + \nu h(x)$$

If the problem is regular (we will define it later) and the point x^* is the local minimum of the problem described above, then there exists ν^* :

Necessary conditions:

$$\nabla_x L(x^*, \nu^*) = 0 \quad \text{that's written above}$$

$$\nabla_\nu L(x^*, \nu^*) = 0 \quad \text{budget constraint}$$

Sufficient conditions:

$$\langle y, \nabla_{xx}^2 L(x^*, \nu^*)y \rangle > 0,$$

$$\forall y \neq 0 \in R^n : \nabla h(x^*)^\top y = 0$$

0.62 Теорема Каруша - Куна - Таккера в форме необходимых условий решения задачи математического программирования.

$$\begin{cases} f_0(x) \rightarrow \min_{x \in \mathbb{R}^d} \\ f_i(x) \leq 0, \quad i = 1, \dots, m, \\ h_i(x) = 0, \quad i = 1, \dots, p. \end{cases} \quad (1)$$

Функция Лагранжа:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x),$$

Пусть x_* — решение задачи. Тогда найдутся такие векторы λ^* и ν^* , что выполнены условия (ККТ).

$$\begin{cases} \nabla f_0(x_*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x_*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x_*) = 0 \\ f_i(x_*) \leq 0, \quad i = 1, \dots, m \\ h_i(x_*) = 0, \quad i = 1, \dots, p \\ \lambda_i^* \geq 0, \quad i = 1, \dots, m \\ \lambda_i^* f_i(x_*) = 0, \quad i = 1, \dots, m \end{cases} \quad (\text{ККТ})$$

Если задача (1) является выпуклой и удовлетворяет условию Слейтера, то условия Куна-Таккера становятся *необходимыми и достаточными*.

0.63 Условие Слейтера.

1. Если задача выпуклая (т.е., говоря о задаче минимизации, оптимизируемая функция f_0 и ограничения вида неравенство f_i — выпуклые, ограничения вида равенства h_i — аффинные)
2. И существует точка x такая, что $h(x) = 0$ и $f_i(x) < 0$ (ограничения вида равенства активные, а ограничения вида неравенства выполняются строго)

То тогда задача имеет нулевой зазор двойственности и условия ККТ становятся необходимыми и достаточными.

0.64 Задача выпуклого программирования

Задача выпуклого программирования — это задача оптимизации, в которой целевая функция является выпуклой функцией и область допустимых решений выпукла.

Говорят, что задача выпуклого программирования представлена в стандартной форме, если она записана как
Минимизировать

$$f_0(x) \rightarrow \min_x$$

При условиях

$$f_i(x) \leq 0, \quad i = 1, \dots, m$$

$$h_i(x) = 0, \quad i = 1, \dots, p,$$

где $x \in \mathbb{R}^n$ является переменной оптимизации, функции f_0, f_1, \dots, f_m выпуклы, а функции h_1, \dots, h_p аффинны.

0.65 Двойственная функция в задаче математического программирования.

Предположим, что $D = \bigcap_{i=0}^m \text{dom } f_i \cap \bigcap_{i=0}^p \text{dom } h_i$ непустое. Определим двойственную функцию $g : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ как минимум лагранжиана по x : для $\lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p$

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) = \inf_{x \in D} f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

Так как двойственная функция это поточечный инфинум семейства аффинных функций от (λ, ν) , она вогнутая, даже если изначальная задача не выпуклая.

0.66 Двойственная задача для задачи математического программирования.

Пусть p^* - оптимальное решение изначальной задачи. Пусть \hat{x} достижимая точка для изначальной задачи, т.е. $f_i(\hat{x}) \leq 0$ and $h_i(\hat{x}) = 0, \lambda \geq 0$. Тогда имеем:

$$L(\hat{x}, \lambda, \nu) = f_0(\hat{x}) + \underbrace{\lambda^T f(\hat{x})}_{\leq 0} + \underbrace{\nu^T h(\hat{x})}_{=0} \leq f_0(\hat{x})$$

Тогда

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) \leq L(\hat{x}, \lambda, \nu) \leq f_0(\hat{x})$$

$$g(\lambda, \nu) \leq p^*$$

Двойственной задачей называется

$$g(\lambda, \nu) \rightarrow \max_{\lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p}$$

$$s.t. \lambda \geq 0$$

0.67 Сильная двойственность. Зазор двойственности.

Пусть p^* - решение прямой задачи, d^* - решение двойственной задачи. Зазором двойственности называется

$$p^* - d^* \geq 0$$

Сильная двойственность возникает, если зазор равен нулю

$$p^* = d^*$$

0.68 Локальный анализ чувствительности с помощью множителей Лагранжа.

Перейдем к возмущенной версии задачи:

$$f_0(x) \rightarrow \min_x$$

$$f_i(x) \leq u_i, \quad i = 1, \dots, m$$

$$h_i(x) = v_i, \quad i = 1, \dots, p,$$

Обозначим $p^*(u, v)$ - оптимальное решение этой задачи. Если имеет место сильная двойственность, то выполнено:

$$p^*(u, v) \geq p^*(0, 0) - \lambda^{*T} u - \nu^{*T} v$$

Если множители Лагранжа λ_i^*, ν_i^* большие, то небольшое изменение ограничений приведет к существенному изменению оптимального решения. То есть соответствующие ограничения очень сильно влияют на задачу.

Если множители Лагранжа маленькие, то соответствующие ограничения мало влияют на задачу.

0.69 Задача линейного программирования. Задача линейного программирования в стандартной форме.

Все задачи с линейным функционалом и линейными ограничениями считаются задачами линейного программирования. Стандартная форма:

$$\min_{x \in \mathbb{R}^n} c^T x$$

$$s.t. Ax = b$$

$$x_i \geq 0, i = 1, \dots, n$$

0.70 Возможные случаи двойственности в задаче линейного программирования.

Двойственная задача:

$$\max_{\nu \in \mathbb{R}^m} -b^T \nu$$

$$s.t. -A^T \nu \leq c$$

- 1) Если либо у прямой, либо у двойственной задачи есть конечное решение, то и у другой тоже, и целевые переменные равны.
- 2) Если либо прямая, либо двойственная задача неограничена, то вторая из них невыполнима.

0.71 Симплекс метод.

Симплекс метод решает следующую задачу:

$$\min_{x \in \mathbb{R}^n} c^\top x$$

$$s.t. Ax \leq b$$

Шаги выполнения симплекс метода:

1. Поиск начальной базисной допустимой точки:

- Выберем начальную базисную (она является решением системы $A_B x = b_B$, где B - базис размера n пространства, а матрица A обычно имеет больше n ограничений) допустимую ($Ax_0 \leq b$) точку x_0 (искать ее будем через двухфазный симплексметод). Если такая точка не найдена, задача не имеет допустимого решения.

2. Проверка оптимальности:

- Разложение вектора c в данном базисе B с коэффициентами λ_B :

$$\lambda_B^\top A_B = c^\top \quad \text{или} \quad \lambda_B^\top = c^\top A_B^{-1}$$

- Если все компоненты λ_B неположительны, текущий базис является оптимальным. Иначе далее меняем вершину симплекса.

3. Определение переменной для удаления из базиса:

- Если в разложении λ_B есть положительные координаты, продолжаем оптимизацию. Пусть $\lambda_B^k > 0$. Необходимо исключить k из базиса. Рассчитаем направляющий вектор d , идя вдоль которого изменим вершину следующим образом: во-первых, для вектором всех ограничений из базиса, которые мы оставляем, направление должно быть им ортогонально, и, во-вторых, вдоль него значение, связанное с нашим ограничением, должно убывать:

$$\begin{cases} A_{B \setminus \{k\}} d = 0 \\ a_k^\top d < 0 \end{cases}$$

4. Вычисление шага вдоль выбранного направления d :

- Для всех $j \notin B$ считаем шаг:

$$\mu_j = \frac{b_j - a_j^\top x_B}{a_j^\top d}$$

- Новая вершина, которую добавим в базис:

$$t = \arg \min_j \{\mu_j \mid \mu_j > 0\}$$

5. Обновление базиса:

- Обновляем базис и текущее решение:

$$B' = B \setminus \{k\} \cup \{t\},$$

$$x_{B'} = x_B + \mu_t d = A_{B'}^{-1} b_{B'}$$

- Изменение базиса приводит к уменьшению значения целевой функции:

$$c^\top x_{B'} = c^\top (x_B + \mu_t d) = c^\top x_B + \mu_t c^\top d$$

6. Повторение:

- Далее повторяем шаги 2-5 до достижения оптимального решения или установления, что задача не имеет допустимого решения.

0.72 Нахождение первоначальной угловой точки с помощью двухфазного симплекс метода.

1. Рассмотрим задачу (Phase 1):

$$\begin{aligned} & \min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \sum_{i=1}^m \xi_i \\ & \text{s.t. } Ay - Az \leq b + \xi \\ & \quad y \geq 0, x \geq 0, \xi \geq 0 \end{aligned}$$

Для нее есть допустимая угловая точка $z = 0, y = 0, \xi_i = \max(0, -b_i)$. Начиная с нее, решим задачу симплекс методом и получим точку оптимума, в которой $\xi = 0$ и выполнены указанные ограничения.

2. Решение задачи Phase 1 является допустимым базисом задачи Phase 2:

$$\begin{aligned} & \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\ & \text{s.t. } Ay - Az \leq b \\ & \quad y \geq 0, x \geq 0 \end{aligned}$$

3. Заметим, что оно так же будет являться допустимым базисом и угловой точкой для исходной задачи:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x \\ & \text{s.t. } Ax \leq b \end{aligned}$$

4. Так и нашли первоначальную угловую точку для исходной задачи.

0.73 Сходимость симплекс метода.

В худшем случае симплекс метод сходится экспоненциально от размерности задачи, но на практике в среднем алгоритм работает сильно лучше. Задача, на которой симплекс метод работает экспоненциальное время, называется примером Klee Minty.

0.74 Показать, что направление антиградиента - направление наискорейшего локального убывания функции.

Рассмотрим линейное приближение дифференцируемой функции f вдоль направления h , $\|h\|_2 = 1$:

$$f(x + \alpha h) = f(x) + \alpha \langle f'(x), h \rangle + o(\alpha)$$

Мы хотим, чтобы h было направлением убывания:

$$f(x + \alpha h) < f(x)$$

$$f(x) + \alpha \langle f'(x), h \rangle + o(\alpha) < f(x)$$

и, переходя к пределу при $\alpha \rightarrow 0$:

$$\langle f'(x), h \rangle \leq 0$$

Также, из неравенства Коши — Буняковского — Шварца:

$$|\langle f'(x), h \rangle| \leq \|f'(x)\|_2 \|h\|_2$$

$$\langle f'(x), h \rangle \geq -\|f'(x)\|_2 \|h\|_2 = -\|f'(x)\|_2$$

Таким образом, направление антиградиента

$$h = -\frac{f'(x)}{\|f'(x)\|_2}$$

даёт направление наискорейшего локального убывания функции f .

0.75 Дифференциальное уравнение градиентного потока.

Дифференциальное уравнение градиентного потока - дифференциальное уравнение, дискретизацией которого является метод градиентного спуска. Выглядит оно следующим образом:

$$\frac{dx}{dt} = -f'(x(t))$$

0.76 Метод градиентного спуска.

Метод градиентного спуска - алгоритм оптимизации функции при помощи изменения аргумента в соответствии с градиентом этой функции. Формально: для поиска минимального значения функции $f(x)$ строим последовательность x_k так, что:

$$x_{k+1} = x_k - \alpha f'(x_k)$$

, где α - размер шага обучения (learning rate).

0.77 Наискорейший спуск.

Скорость сходимости метода градиентного спуска напрямую зависит от выбора α , наискорейший спуск достигается при выборе оптимального шага на каждой итерации алгоритма:

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Оптимальное условие:

$$\nabla f(x_{k+1})^\top \nabla f(x_k) = 0$$

0.78 Липшицева парабола для гладкой функции.

Формула для Липшицевой параболы для гладкой функции, подпирающей исходную функцию $f(x)$ сверху в каждой точке:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2} \|x^{k+1} - x^k\|^2$$

0.79 Размер шага наискорейшего спуска для квадратичной функции.

Пусть $f(x) = x^\top Ax - b^\top x + c$ - квадратичная задача. Тогда для нее оптимальный шаг:

$$\alpha^* = \frac{2}{\mu + L}$$

где μ, L - наименьшее и наибольшее собственные значения матрицы A соответственно.

0.80 Характер сходимости градиентного спуска к локальному экстремуму для гладких невыпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$\|\nabla f(x_k)\|^2 \sim \mathcal{O}\left(\frac{1}{k}\right).$$

0.81 Характер сходимости градиентного спуска для гладких выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$$

0.82 Характер сходимости градиентного спуска для гладких и сильно выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$\|x_k - x^*\|^2 \sim \mathcal{O}\left((1 - \frac{\mu}{L})^k\right)$$

0.83 Связь спектра гессиана с константами сильной выпуклости и гладкости функции.

$$\mu = \min_{x \in \text{dom } f} \lambda_{\min}(\nabla^2 f(x)), \quad L = \max_{x \in \text{dom } f} \lambda_{\max}(\nabla^2 f(x)).$$

0.84 Условие Поляка-Лоясиевича (градиентного доминирования) для функций.

Функцию f называют PL функцией (с константой μ), если:

$$\exists \mu > 0 : \|\nabla f(x)\|^2 \geq 2\mu(f(x) - f^*) \quad \forall x$$

где f^* – минимальное значение функции.

Градиентный спуск для таких функций сходится линейно. Функция может не быть выпуклой, но быть PL функцией.

0.85 Сходимость градиентного спуска для сильно выпуклых квадратичных функций.

Оптимальные гиперпараметры.

Решаем задачу минимизации методом градиентного спуска. Пусть $A \in \mathbb{S}_{++}^n \Rightarrow \nabla f = Ax - b$.

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \rightarrow \min_{x \in \mathbb{R}^d}$$

$$x_{k+1} = x_k - \alpha(Ax_k - b)$$

$$\alpha_{opt} = \frac{2}{\mu + L}, \text{ где } \mu = \lambda_{\min}(A), L = \lambda_{\max}(A)$$

$$\kappa = \frac{L}{\mu} \geq 1$$

$$\rho = \frac{\kappa - 1}{\kappa + 1}$$

$$\|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|$$

0.86 Связь PL-функций и сильно выпуклых функций.

Любая дифференцируемая и μ -сильно выпуклая $f(x)$ является PL функцией с константой μ . (Обратное неверно)

0.87 Привести пример выпуклой, но не сильно выпуклой задачи линейных наименьших квадратов (возможно, с регуляризацией).

Рассмотрим задачу минимизации функции:

$$\|Ax - b\|_2^2 \rightarrow \min_{x \in \mathbb{R}^n},$$

где матрица $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m < n$ (лежащая).

0.88 Привести пример сильно выпуклой задачи линейных наименьших квадратов (возможно, с регуляризацией).

Рассмотрим задачу минимизации функции:

$$f(x) = \|Ax - b\|_2^2,$$

где $A \in \mathbb{R}^{n \times n}$ (ранг $A = n$). Эта функция сильно выпукла, так как гессиан положительно определен.

0.89 Привести пример выпуклой негладкой задачи линейных наименьших квадратов (возможно, с регуляризацией).

Рассмотрим задачу минимизации функции:

$$f(x) = \|Ax - b\|_2^2 + \lambda \|x\|_1,$$

где $A \in \mathbb{R}^{n \times n}$, $\lambda > 0$. Эта функция выпукла, но негладка из-за наличия ℓ_1 -регуляризации.

0.90 Субградиент. Субдифференциал.

Субградиент функции $f(x)$ в точке x_0 — это вектор g , удовлетворяющий условию:

$$f(x) \geq f(x_0) + g^T(x - x_0), \quad \forall x$$

Градиент — частный случай субградиента.

Множество всех субградиентов в точке x_0 называется субдифференциалом и обозначается как $\partial f(x_0)$.

0.91 Субградиентный метод.

Пусть g_k — субградиент $f(x)$ в точке x_k . Шаг метода:

$$x_{k+1} = x_k - \alpha g_k$$

0.92 Характер сходимости субградиентного метода для негладких выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$f(x_k) - f^* \in \mathcal{O}\left(1/\sqrt{k}\right).$$

0.93 Нижние оценки для гладкой выпуклой оптимизации с помощью методов первого порядка в терминах \mathcal{O} от числа итераций метода.

$$\mathcal{O}(1/k^2).$$

0.94 Отличие ускоренной и неускоренной линейной сходимости для методов первого порядка.

Тип	Неускоренная	Ускоренная
Гладкая и сильно-выпуклая (или PL)	$\mathcal{O}\left((1 - \frac{\mu}{L})^k\right)$	$\mathcal{O}\left((1 - \sqrt{\frac{\mu}{L}})^k\right)$

0.95 Метод тяжелого шарика (Поляка).

$$x_{k+1} = x_k - \alpha \nabla f(x) + \beta(x_k - x_{k-1})$$

0.96 Ускоренный градиентный метод Нестерова для выпуклых гладких функций.

Рассматриваем задачу $f(x) \rightarrow \min$, где $f : \mathbb{R}^n \rightarrow \mathbb{R}$ выпуклая и L -гладкая. Алгоритм Нестерова ускоренного градиентного спуска (NAG) имеет вид ($x_0 = y_0, \lambda_0 = 0$):

Обновление градиента:

$$y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$$

Экстраполяция:

$$x_{k+1} = (1 - \gamma_k)y_{k+1} + \gamma_k y_k$$

Экстраполяция веса:

$$\lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2}$$

Экстраполяция веса:

$$\gamma_k = \frac{1 - \lambda_k}{\lambda_{k+1}}$$

Метод сходится со скоростью $O\left(\frac{1}{k^2}\right)$, а именно:

$$f(y_k) - f^* \leq \frac{2L\|x_0 - x^*\|_2^2}{k^2}$$

0.97 Ускоренный градиентный метод Нестерова для сильно выпуклых гладких функций.

Рассматриваем задачу $f(x) \rightarrow \min$, где $f : \mathbb{R}^n \rightarrow \mathbb{R}$ — μ -сильно выпуклая и L -гладкая. Алгоритм Нестерова ускоренного градиентного спуска (NAG) имеет вид ($x_0 = y_0, \lambda_0 = 0$):

Обновление градиента:

$$y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$$

Экстраполяция:

$$x_{k+1} = (1 - \gamma_k)y_{k+1} + \gamma_k y_k$$

Экстраполяция весов:

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

Метод сходится линейно, а именно:

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

0.98 Проекция.

$$\text{proj}_S(y) = \frac{1}{2} \arg \min_{x \in S} \|x - y\|_2^2$$

0.99 Достаточное условие существования проекции точки на множество.

S — замкнутое

0.100 Достаточное условие единственности проекции точки на множество.

S — замкнутое и выпуклое

0.101 Метод проекции градиента.

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

0.102 Критерий проекции точки на выпуклое множество (Неравенство Бурбаки-Чейни-Гольдштейна).

Пусть $S \subseteq \mathbb{R}^n$ замкнутое и выпуклое множество, $\forall x \in S, y \in \mathbb{R}^n$. Тогда

$$\langle y - \text{proj}_S(y), x - \text{proj}_S(y) \rangle \leq 0$$

$$\|x - \text{proj}_S(y)\|^2 + \|y - \text{proj}_S(y)\|^2 \leq \|x - y\|^2$$

0.103 Проекция как нерастягивающий оператор.

$$\|\text{proj}(x) - \text{proj}(y)\|_2 \leq \|x - y\|_2.$$

0.104 Метод Франк-Вульфа.

$$y_k = \arg \min_{x \in S} f_{x_k}^I(x) = \arg \min_{x \in S} \langle \nabla f(x_k), x \rangle$$

$$x_{k+1} = \gamma_k x_k + (1 - \gamma_k) y_k$$

0.105 Характер сходимости метода проекции градиента для гладких выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$O(\frac{1}{k})$$

0.106 Характер сходимости метода проекции градиента для гладких сильно выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$O((1 - \frac{\mu}{L})^k)$$

0.107 Характер сходимости метода Франк-Вульфа для гладких выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$O(\frac{1}{k})$$

0.108 Характер сходимости метода Франк-Вульфа для гладких сильно выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$O(\frac{1}{k^2})$$

0.109 A-сопряженность двух векторов. A-ортогональность. Скалярное произведение

$$\langle \cdot, \cdot \rangle_A$$

A - симметричная, положительно определённая матрица

A -сопряженность двух векторов. A -ортогональность: $x^T A y = 0$

Скалярное произведение $\langle x, y \rangle_A = x^T A y$

0.110 Процедура ортогонализации Грама-Шмидта.

Вход: n линейно независимых векторов u_0, \dots, u_{n-1} .

Выход: n линейно независимых векторов, которые попарно ортогональны d_0, \dots, d_{n-1} .

$$d_k = u_k + \sum_{i=0}^{k-1} \beta_{ik} d_i \quad \beta_{ik} = -\frac{\langle d_i, u_k \rangle}{\langle d_i, d_i \rangle}$$

0.111 Метод сопряженных направлений.

Рассматриваем задачу:

$$f(x) = \frac{1}{2} x^T A x - b^T x + c \rightarrow \min_{x \in \mathbb{R}^d}$$

Идея:

- В изотропном $A = I$ мире, наискорейший спуск стартующий из произвольной точки в любом пространстве, натянутом на линейную оболочку из n ортогональных линейно независимых векторов, будет сходиться за n шагов в точной арифметике. Мы попытаемся в случае $A \neq I$ провести A-ортогонализацию, чтобы "наискорейшим" образом спускаться в измененном базисе.
- Предположим, имеется набор из n линейно независимых A-ортогональных векторов (направлений) d_0, \dots, d_{n-1} (которые, например, были получены в ходе A-ортогонализации Г-III).
- Мы хотим создать метод, который переходит от x_0 к x^* по указанным ортогональным $d - 1$ направлениям с некоторыми шагами, т.е. $x_0 - x^* = \sum_{i=0}^{d-1} \alpha_i d_i$, где α_i - из решения задачи линейного поиска.

Алгоритм

- $k = 0$

$$x_k = x_0$$

$$d_k = d_0 = -\nabla f(x_0).$$

- Пока $k < n$ (строим n направлений):
 - Линейный поиск шага: хотим найти $\alpha : f(x_k + \alpha_k d_k) \rightarrow \min$

$$\alpha = -\frac{d_k^T (Ax - b)}{d_k^T Ad_k}$$

— Шаг алгоритма:

$$x_{k+1} = x_k + \alpha_k d_k$$

— Обновляем направление: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ чтобы сделать $d_{k+1} \perp_A d_k$, где β_k считается по формуле:

$$\beta_k = \frac{\nabla f(x_{k+1})^\top A d_k}{d_k^\top A d_k}.$$

— $k = k + 1$

0.112 Метод сопряженных градиентов.

Рассматриваем задачу:

$$f(x) = \frac{1}{2} x^\top A x - b^\top x + c \rightarrow \min_{x \in \mathbb{R}^d}$$

Conjugate gradients method

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$$

if \mathbf{r}_0 is sufficiently small, then return \mathbf{x}_0 as the result

$$\mathbf{d}_0 := \mathbf{r}_0$$

$$k := 0$$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^\top \mathbf{r}_k}{\mathbf{d}_k^\top \mathbf{A} \mathbf{d}_k}$$

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

$$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{d}_k$$

if \mathbf{r}_{k+1} is sufficiently small, then exit loop

$$\beta_k := \frac{\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}}{\mathbf{r}_k^\top \mathbf{r}_k}$$

$$\mathbf{d}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$$

$$k := k + 1$$

end repeat

return \mathbf{x}_{k+1} as the result

0.113 Зависимость сходимости метода сопряженных градиентов от спектра матрицы.

Если матрица A имеет только r различных собственных чисел, тогда метод сопряжённых градиентов сходится за r итераций.

0.114 Характер сходимости метода сопряженных градиентов в терминах \mathcal{O} от числа итераций метода.

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A,$$

где $\|x\|_A^2 = x^\top Ax$

$\kappa(A) = \frac{\lambda_1(A)}{\lambda_n(A)}$ – число обусловленности

$A, \lambda_1(A) \geq \dots \geq \lambda_n(A)$ – собственные числа матрицы A

Получаем следующую оценку на число итераций при заданной точности ε : $\|x_k - x^*\|_A \leq \|x_0 - x^*\|_A$

$$k \leq \left\lceil \frac{1}{2} \sqrt{\kappa(A)} \ln \left(\frac{2}{\varepsilon} \right) \right\rceil$$

0.115 Метод Поляка-Рибьера.

!!! Кажется этого не было на лекциях, взяла описание из другого труда !!!

Используется для минимизации неквадратичных выпуклых функций. Без знания аналитического выражения шаг 2 алгоритма метода сопряжённых направлений вместо подсчёта α из минимизации $f(x_k + \alpha_k d_k)$ находим α обычным линейным поиском.

$$\beta_k = \frac{\nabla f(x_{k+1})^\top A(\nabla f(x_{k+1}) - \nabla f(x_k))}{d_k^\top A(\nabla f(x_{k+1}) - \nabla f(x_k))}.$$

0.116 Метод Ньютона.

Рассматривается задача минимизации функции с невырожденным гессианом.

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

0.117 Сходимость метода Ньютона для квадратичной функции.

Метод Ньютона сходится для квадратичной функции за одну итерацию. Следует из метода Ньютона квадратичной тейлоровской аппроксимации:

$$f(x) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2} (x - x_k)^\top \nabla^2 f(x_k) (x - x_k), \quad \nabla f(x_{k+1}) = 0$$

0.118 Характер сходимости метода Ньютона для сильно выпуклых гладких функций – куда и как сходится.

Пусть $f(x)$ – сильно выпукла дважды непрерывно дифференцируемая на \mathbb{R}^n , и выполняются неравенства: $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$. Тогда метод Ньютона с постоянным шагом локально сходится к решению со сверхлинейной

скоростью. Если в добавок Гессиан M -Липшицев тогда метод сходится локально к x^* с квадратичной скоростью.

0.119 Демпфированный метод Ньютона.

$$x_{k+1} = x_k - \alpha_k [\nabla^2 f(x_k)]^{-1} \nabla f(x_k), \quad \alpha_k \in [0, 1]$$

где α_k находят с помощью линейного поиска. Сходимость глобальная.

0.120 Идея квазиньютоновских методов. Метод SR-1.

Идея

Для классической задачи безусловной оптимизации $f(x) \rightarrow \min_{x \in \mathbb{R}^n}$ общий вид итерации может быть записан как:

$$x_{k+1} = x_k + \alpha_k d_k$$

В методе Ньютона направление d_k задается решением системы линейных уравнений на каждом шаге:

$$B_k d_k = -\nabla f(x_k), \quad B_k = \nabla^2 f(x_k)$$

$$d_k = B^{-1} \nabla f(x_k)$$

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

т.е. на каждой итерации необходимо вычислять гессиан и градиент и решать линейную систему (это плохо и дорого).

Заметим, что если для каждого шага будем брать $B_k = I_n$, мы получим в точности метод градиентного спуска. Общая схема квазиньютоновских методов основана на выборе B_k , которая в каком-то смысле сходится при $k \rightarrow \infty$ к настоящему значению гессиана $\nabla^2 f(x_k)$.

Шаблон квазиньютоновского метода

Пусть $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. Для $k = 1, 2, 3, \dots$, повторим:

1. Решить $B_k d_k = -\nabla f(x_k)$ относительно d_k
2. Обновить $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить B_{k+1} из B_k

SR-1

$$B_{k+1} = B_k + \frac{(\Delta y_k - B_k d_k)(\Delta y_k - B_k d_k)^T}{(\Delta y_k - B_k d_k)^T d_k}$$

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

0.121 Нижние оценки для негладкой выпуклой оптимизации с помощью методов первого порядка в терминах \mathcal{O} от числа итераций метода.

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$

- Subgradient method is optimal for the problems above.
- One can use Mirror Descent (a generalization of the subgradient method to a possibly non-Euclidian distance) with the same convergence rate to better fit the geometry of the problem.
- However, we can achieve standard gradient descent rate $\mathcal{O}\left(\frac{1}{k}\right)$ (and even accelerated version $\mathcal{O}\left(\frac{1}{k^2}\right)$) if we will exploit the structure of the problem.

0.122 Проксимальный оператор.

$$\text{prox}_{f,\alpha}(x_k) = \arg \min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]$$

Ещё одна форма записи:

$$\text{prox}_{f,\alpha}(x_k) = \text{prox}_{\alpha f}(x_k) = \arg \min_{x \in \mathbb{R}^n} \left[\alpha f(x) + \frac{1}{2} \|x - x_k\|_2^2 \right]$$

0.123 Оператор проекции как частный случай проксимального оператора.

Пусть \mathbb{I}_S –индикатор для закрытого, выпуклого множества (0 , если $x \in S$, иначе ∞). Тогда ортогональная проекция:

$$\pi_S(y) = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - y\|_2^2 = \arg \min_{x \in S} \left[\frac{1}{2} \|x - y\|_2^2 + \mathbb{I}_S(x) \right]$$

Получается, что для получения проксимального оператора достаточно заменить \mathbb{I}_S на какую-либо выпуклую функцию $r(x)$:

$$\text{prox}_r(y) = \text{prox}_{r,1}(y) = \arg \min_{x \in \mathbb{R}^n} \left[r(x) + \frac{1}{2} \|x - y\|_2^2 \right]$$

0.124 Характер сходимости проксимального градиентного метода для гладких выпуклых функций f в терминах \mathcal{O} от числа итераций метода.

Сходимость: $\mathcal{O}\left(\frac{1}{k}\right)$.

Theorem

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}(x_k - \alpha \nabla f(x_k))$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \arg \min_{x \in \mathbb{R}^n} [\alpha r(x) + \frac{1}{2} \|x - x_k\|_2^2]$ can be evaluated.

Proximal gradient descent with fixed step size $\alpha = 1/L$ satisfies

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

скоро 76 GD ✓

Proximal gradient descent has a convergence rate of $O(1/k)$ or $O(1/\varepsilon)$. This matches the gradient descent rate! (But remember the proximal operation cost)

0.125 Характер сходимости проксимального градиентного метода для гладких сильно выпуклых функций f в терминах \mathcal{O} от числа итераций метода.

Сходимость: $\mathcal{O}(e^{-k})$.

Theorem

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}(x_k - \alpha \nabla f(x_k))$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is μ -strongly convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \arg \min_{x \in \mathbb{R}^n} [\alpha r(x) + \frac{1}{2} \|x - x_k\|_2^2]$ can be evaluated.

Proximal gradient descent with fixed step size $\alpha \leq 1/L$ satisfies

$$\|x_{k+1} - x^*\|_2^2 \leq (1 - \alpha\mu)^k \|x_0 - x^*\|_2^2$$

This is exactly gradient descent convergence rate. Note, that the original problem is even non-smooth!

0.126 Аналитическое выражение для $\text{prox}_{\lambda \|x\|_1}$.

$$\text{prox}_{\lambda \|x\|_1}(x) = \arg \min_{y \in \mathbb{R}^n} \left[\lambda \|y\|_1 + \frac{1}{2} \|x - y\|_2^2 \right] = [|x| - \lambda]_+ \cdot \text{sign}(x)$$

Иначе это ещё называют soft thresholding operator.

0.127 Аналитическое выражение для $\text{prox}_{\frac{\mu}{2} \|x\|_2^2}$.

$$\text{prox}_{\frac{\mu}{2} \|x\|_2^2}(x) = \frac{x}{1 + \mu}$$

0.128 Проксимальный оператор как нерастягивающий оператор.

Проксимальный оператор является нерастягивающим:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \|x - y\|_2^2$$

А также является строго нерастягивающим:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \langle \text{prox}_r(x) - \text{prox}_r(y), x - y \rangle$$

0.129 Характер сходимости ускоренного проксимального градиентного метода для гладких выпуклых функций f в терминах \mathcal{O} от числа итераций метода.

Сходимость: $\mathcal{O}\left(\frac{1}{k^2}\right)$.

Accelerated Proximal Method

Let $x_0 = y_0 \in \text{dom}(r)$. For $k \geq 1$:

$$\begin{aligned} x_k &= \text{prox}_{\alpha_k h}(y_{k-1} - \alpha_k \nabla f(y_{k-1})) \\ y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1}) \end{aligned}$$

Achieves

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

0.130 Метод стохастического градиентного спуска.

Хотим $f(x) \rightarrow \min_{x \in \mathbb{R}^p}$, где $f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$

SGD:

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x)$$

Где i_k - случайный индекс. При $P(i_k = i) = \frac{1}{n}$, $\mathbb{E}[\nabla f_{i_k}(x)] = \nabla f(x)$

0.131 Идея мини-батча для метода стохастического градиентного спуска. Эпоха.

Разделим данные размера N на k мини-батчей (выборок) размера \mathcal{B}_k , на каждой итерации посчитаем градиент мини-батча (можно параллельно). Эпоха - k итераций с батчем размера $\mathcal{B}_k = \frac{N}{k}$

Один шаг спуска

$$x_{k+1} = x_k - \alpha_k \frac{1}{\|\mathcal{B}_k\|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x)$$

С увеличением размера мини-батча время на эпоху уменьшается до тех пор, пока батч влезает в память

0.132 Характер сходимости стохастического градиентного спуска для гладких выпуклых функций в терминах \mathcal{O} от числа итераций метода.

$$f \text{ - гладкая выпуклая} \Rightarrow k \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$$

где ε - точность, до которой мы хотим сойтись

$$\text{От числа итераций: } \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

0.133 Характер сходимости стохастического градиентного спуска для гладких PL-функций в терминах \mathcal{O} от числа итераций метода.

$$f \text{ - гладкая PL} \Rightarrow k \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$$

$$\text{От числа итераций: } \mathcal{O}\left(\frac{1}{k}\right)$$

0.134 Характер работы стохастического градиентного спуска с постоянным шагом для гладких PL-функций.

$$\mathbb{E}[f(x_{k+1})] - f^* \leq (1 - 2\alpha\mu)[f(x_k) - f^*] + \frac{L\sigma^2\alpha}{4\mu}$$

где α - const

0.135 Основная идея методов уменьшения дисперсии.

Хотим уменьшить дисперсию случайной величины X . Пусть Y другая с.в. с известным мат ожиданием. Рассмотрим с.в. $Z_\alpha = \alpha(X - Y) + \mathbb{E}[Y]$. Тогда:

$$\mathbb{E}[Z_\alpha] = \alpha\mathbb{E}[X] + (1 - \alpha)\mathbb{E}[Y]$$

$$\text{var}(Z_\alpha) = \alpha^2(\text{var}(X) + \text{var}(Y) - 2\text{cov}(X, Y))$$

При $\alpha = 1$ - нет смещения мат ожидания. При уменьшении α матожидание смещается, но уменьшается дисперсия (в случае, если Y хорошо скоррелирован с X).

0.136 Метод SVRG (Stochastic Variance Reduced Gradient)

- Инициализируем \tilde{x}
- for i in epoch number
 - считаем полный градиент $f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$
 - $x_0 = \tilde{x}$
 - for t in epoch length (m - гиперпараметр):
 - j - случайный индекс

$$x_t = x_{t-1} - \alpha[\nabla f(\tilde{x}) + \nabla f_j(x_{t-1}) - \nabla f_j(\tilde{x})]$$

— Обновляем веса $\hat{x} = x_m$

Таким образом: используем метод уменьшения дисперсии, но долгие итерации (на каждой эпохе пересчитываем полный градиент + 2 градиента во внутреннем цикле)

0.137 Метод SAG (Stochastic average gradient)

- Инициализируем $x^{(0)}$, $g_i^0 = \nabla f_i(x^{(0)})$. И храним всю таблицу градиентов!
- На каждом шаге выбираем случайный индекс $i_k \in \{1 \dots n\}$
- $g_{i_k}^{(k)} = \nabla f_{i_k}(x^{(k-1)})$
- остальные градиенты оставляем без изменений $g_i^{(k)} = g_i^{(k-1)}$
- $x^{(k)} = x^{(k)} - \alpha^k g^{(k)} = x^{(k)} - \alpha^k \frac{1}{n} \sum_{i=1}^n g_i^{(k)}$. Среднее пересчитывается быстро, так как изменилось одно слагаемое (отняли старое значение, добавили новое)

Получили метод со скоростью итерации как в базовом SGD, сходимостью в выпуклом случае $\sim \mathcal{O}\left(\frac{1}{k}\right)$, в сильновыпуклом $\mathcal{O}(\gamma^k)$

Из минусов: не сможем обучить большие модели, потому что нужно хранить огромную таблицу с градиентами

0.138 Метод Adagrad.

$$\begin{aligned} g^{(k)} &= \nabla f_{i_k}(x^{(k-1)}) \\ v_j^{(k)} &= v_j^{(k-1)} + (g_j^{(k)})^2 \\ x_j^{(k)} &= x_j^{(k-1)} - \alpha \frac{g_j^{(k)}}{\sqrt{v_j^{(k)}} + \epsilon} \end{aligned}$$

Если бы убрали знаменатель в последнем слагаемом, то получили бы обычный SGD. ϵ — для исключения деления на ноль на практике.

- AdaGrad does not require tuning the learning rate: $\alpha > 0$ is a fixed constant, and the learning rate decreases naturally over iterations.
- Can drastically improve over SGD in sparse problems.
- The learning rate of rare informative features diminishes slowly
- Main weakness is the monotonic accumulation of gradients in the denominator. AdaDelta, Adam, AMSGrad, etc. improve on this, popular in training deep neural networks

0.139 Метод RMSProp.

В AdaGrad learning rate может слишком быстро падать. Для этого используем скользящее среднее для обновления параметра v

$$\begin{aligned} g^{(k)} &= \nabla f_{i_k}(x^{(k-1)}) \\ v_j^{(k)} &= \gamma v_j^{(k-1)} + (1 - \gamma)(g_j^{(k)})^2 \\ x_j^{(k)} &= x_j^{(k-1)} - \alpha \frac{g_j^{(k)}}{\sqrt{v_j^{(k)}} + \epsilon} \end{aligned}$$

0.140 Метод Adadelta.

RMSProp зависит от гиперпараметра α . Попробуем избавиться от него.

$$\begin{aligned} v_j^{(k)} &= \gamma v_j^{(k-1)} + (1 - \gamma)(g_j^{(k)})^2 \\ \tilde{g}_j^{(k)} &= \frac{\sqrt{\Delta x_j^{(k-1)} + \epsilon}}{\sqrt{v_j^{(k)}} + \epsilon} g_i^{(k)} \\ x_j^{(k)} &= x_j^{(k-1)} - \tilde{g}_j^{(k)} \\ \Delta x_j^{(k)} &= \rho \Delta x_j^{(k-1)} + (1 - \rho)(\tilde{g}_j^{(k)})^2 \end{aligned}$$

0.141 Метод Adam.

- Suitable for large datasets and high-dimentional optimization problems.
- Adam не сходится для выпуклых функций гыгыгы.

Задача: $f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$

$$\begin{aligned} m_j^{(k)} &= \beta_1 m_j^{(k-1)} + (1 - \beta_1) g_j^{(k)} \\ v_j^{(k)} &= \beta_2 v_j^{(k-1)} + (1 - \beta_2)(g_j^{(k)})^2 \\ \tilde{m}_j &= \frac{m_j^{(k)}}{1 - \beta_1^k}, \quad \tilde{v}_j = \frac{v_j^{(k)}}{1 - \beta_2^k} \\ x_j^{(k)} &= x_j^{(k-1)} - \alpha \frac{\tilde{m}_j}{\sqrt{\tilde{v}_j} + \epsilon} \end{aligned}$$

0.142 Идея проекции функции потерь нейронной сети на прямую, плоскость.

Пусть $L(w)$ - функция от $w \in \mathbb{R}^p$. Введем проекцию на линию:

$$L(\alpha) = L(w_0 + \alpha w_1), \quad \alpha \in [-b, b]$$

где w_0 – вектор весов модели, w_1 – случайный вектор, имеющий ту же “структурку”, что и w_0 . Аналогично можно ввести проекцию на плоскость

$$L(\alpha, \beta) = L(w_0 + \alpha w_1 + \beta w_2)$$

где w_1, w_2 – случайные гауссовые векторы (авторы статьи предлагают дополнительно нормализовать w_1, w_2 так, чтобы их координаты были похожи на норму весов проецируемой модели).

- Два случайных вектора большой размерности с высокой вероятностью ортогональны друг другу.
- Если проекция функции невыпукла, то и исходная функция невыпукла. Таким образом можно заглянуть на устройство функции от многих переменных.
- Если в проекции точка находится “на склоне”, то и в многомерном функционале метод не находится в оптимуме.

0.143 Grokking.

Grokking при обучении нейронных сетей — это явление, когда модель после продолжительного обучения сначала демонстрирует плохую обобщающую способность на новых данных, несмотря на хорошее качество на обучающем наборе. Затем, после дальнейшего обучения, модель неожиданно начинает показывать значительно лучшую производительность и на тестовых данных. Это подразумевает, что модель в конечном итоге находит более глубокие и универсальные закономерности, которые позволяют ей лучше обобщать на неизвестные данные.

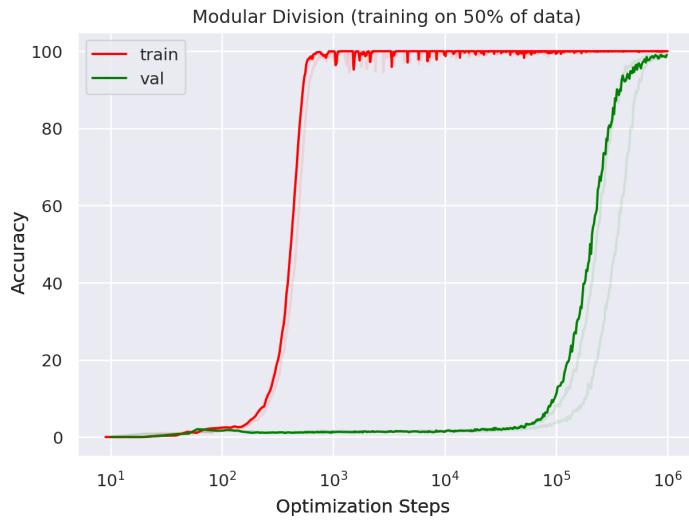


Рис. 1: Греккинг

0.144 Double Descent.

Double descent — это явление, наблюдаемое при обучении нейронных сетей, когда увеличение количества параметров модели сначала приводит к снижению ошибки на обучающем и тестовом наборах (классическое поведение bias-variance tradeoff), затем происходит резкое увеличение ошибки (первая точка перегиба, связанная с переобучением), после чего, с дальнейшим увеличением количества параметров, ошибка снова начинает уменьшаться, формируя вторую “волну” улучшения. Это поведение отличается от традиционной U-образной кривой, и его понимание важно для эффективной настройки гиперпараметров и выбора архитектуры модели.

Рукомахательное объяснение лектора, откуда может возникать Double descent в задачах с полиномами: с одной стороны, слишком большая степень многочлена приводит к overfitting модели на тренировочную выбор-

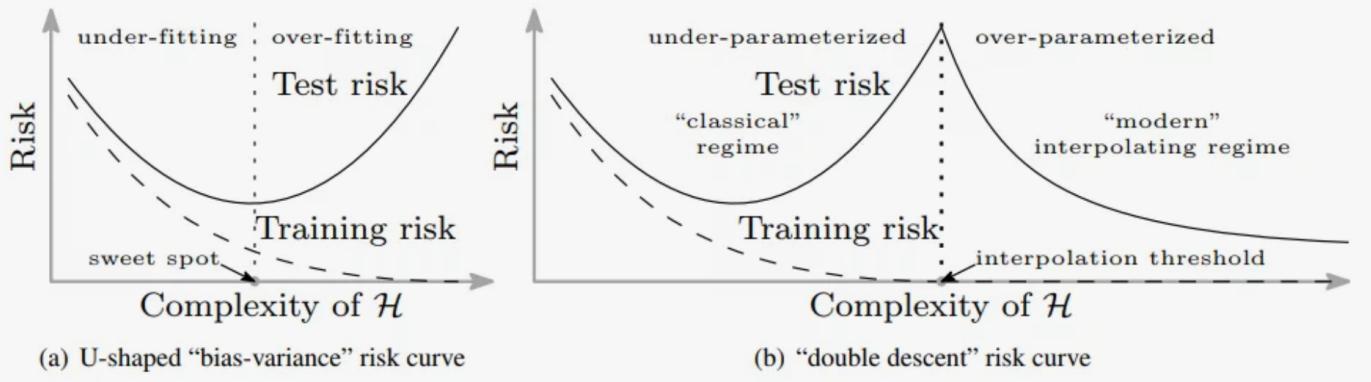


Рис. 2: Иллюстрация зависимости обобщающей способности модели от размера

ку. С другой стороны, существует очень много полиномов, например, 20-й степени, которые с нулевой ошибкой предсказывают обучающую выборку из 7 точек, и некоторые из этих полиномов хорошо предсказывают и тест. Оказывается, что если неким образом регуляризовать модель, то повышается вероятность среди многообразия полиномов большой степени выбрать адекватный, который хорошо предскажет и трейн, и тест.

0.145 Взрыв/Затухание градиентов при обучении глубоких нейронных сетей.

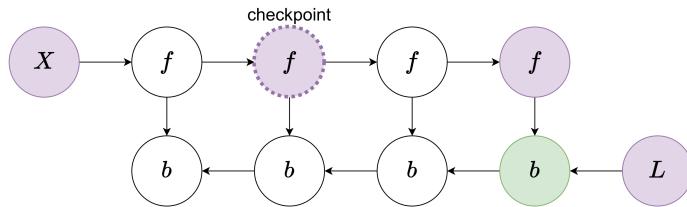
При обучении глубоких нейронных сетей часто возникают проблемы взрыва и затухания градиентов, что приводит к медленной или нестабильной сходимости модели. Эти явления можно описать с помощью производной функции ошибки L по весам сети W . Пусть L - функция потерь, а $\frac{\partial L}{\partial W}$ - градиенты, используемые для обновления весов. Когда сеть имеет много слоев, градиенты вычисляются как произведение матриц Якоби каждого слоя: $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z^{(n)}} \cdot \dots \cdot \frac{\partial z^{(n)}}{\partial z^{(n-1)}} \dots \frac{\partial z^{(2)}}{\partial z^{(1)}}$.

$\frac{\partial z^{(1)}}{\partial W}$, где $z^{(i)}$ – активация i -го слоя. Если значения производных $\frac{\partial z^{(i+1)}}{\partial z^{(i)}}$ в среднем больше единицы, градиенты начинают экспоненциально увеличиваться при обратном распространении, вызывая взрыв градиентов. Напротив, если значения производных меньше единицы, градиенты экспоненциально уменьшаются, что приводит к их затуханию.

0.146 Идея gradient checkpointing.

Gradient checkpointing – это техника, которая позволяет значительно снизить потребление памяти при обучении глубоких нейронных сетей за счет стратегического пересчета промежуточных активаций во время обратного распространения ошибки. В стандартном процессе обучения с использованием обратного распространения ошибка вычисляется для каждого слоя и промежуточные активации сохраняются в памяти, что требует $O(N)$ памяти, где N – количество слоев в сети. При gradient checkpointing вместо сохранения активаций для всех слоев, мы сохраняем их только для некоторых стратегически выбранных слоев, называемых чекпоинтами. Активации для остальных слоев пересчитываются на этапе обратного распространения, что снижает общее потребление памяти. Если мы сохраняем активации через каждые k слоев, то потребление памяти уменьшается до $O(\frac{N}{k})$. Однако, это приводит к

дополнительным вычислительным затратам, так как активации некоторых слоев пересчитываются несколько раз.



0.147 Идея аккумуляции градиентов.

Аккумуляция градиентов — это метод, используемый для эффективного обучения больших нейросетевых моделей, когда ограничен объем доступной видеопамяти. Вместо обновления весов модели после каждого батча данных, как это происходит в стандартном стохастическом градиентном спуске (SGD), градиенты накапливаются в течение нескольких батчей. Затем обновление весов происходит только после накопления градиентов от нескольких батчей, эквивалентных одному большому батчу. Этот подход позволяет использовать меньший объем памяти, так как не требуется хранить большие батчи данных в видеопамяти, при этом достигается сходный с большим батчем эффект на обновление весов, что способствует более стабильному и эффективному обучению модели 3.

0.148 Зачем увеличивать батч при обучении больших нейросетевых моделей. Warmup.

Если увеличивать размер батча, то, при наличии параллелизма, время прохождения эпохи уменьшается. Эмпирическое правило: когда размер минибатча увеличился в k раз, learning rate тоже необходимо увеличить. Для задач типа SGD, SGD+momentum, Heavy ball в k раз (linear scaling rule), а для адаптивных методов (Adam, RMSProp...) эмпирически используется шкалирование базового learning rate в \sqrt{k} раз (square root scaling rule).

Warmup: При этом важно не резко изменять learning rate, а в течение нескольких эпох линейно увеличивать его от старого значения до целевого. Эта техника помогает избежать проблем, связанных с нестабильностью градиентов и резкими изменениями параметров модели в самом начале обучения.

Without gradient accumulation

```

for i, (inputs, targets) in enumerate(data):
    outputs = model(inputs)
    loss = criterion(outputs, targets)
    loss.backward()

    optimizer.step()
    optimizer.zero_grad()
    
```

With gradient accumulation

```

for i, (inputs, targets) in enumerate(data):
    outputs = model(inputs)
    loss = criterion(outputs, targets)
    loss.backward()

    if (i+1) % accumulation_steps == 0:
        optimizer.step()
        optimizer.zero_grad()
    
```

Рис. 3: Идея аккумуляции градиентов

Data Parallel training

1. Parameter server sends the full copy of the model to each device
2. Each device makes forward and backward passes
3. Parameter server gathers gradients
4. Parameter server updates the model

D P

Per device batch size: b . Overall batchsize: Db . Data parallelism involves splitting the data across multiple GPUs, each with a copy of the model. Gradients are averaged and weights updated synchronously:

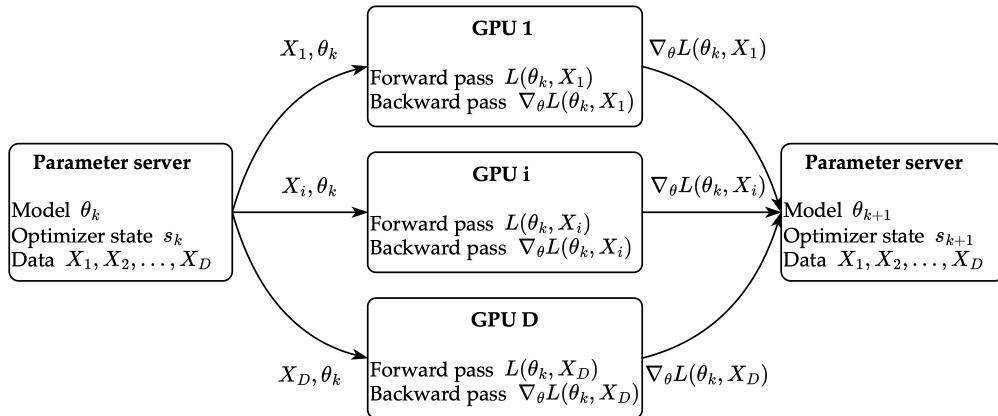


Рис. 4: Data Parallel

0.149 Data Parallel обучение на нескольких видеокартах.

4

0.150 GPipe Pipeline параллелизм.

5

0.151 PipeDream Pipeline параллелизм.

6

0.152 Дообучение больших моделей с помощью LoRA адаптеров.

7

Pipeline model parallelism (GPipe) ⁸

GPipe splits the model into stages, each processed sequentially. Micro-batches are passed through the pipeline, allowing for overlapping computation and communication:

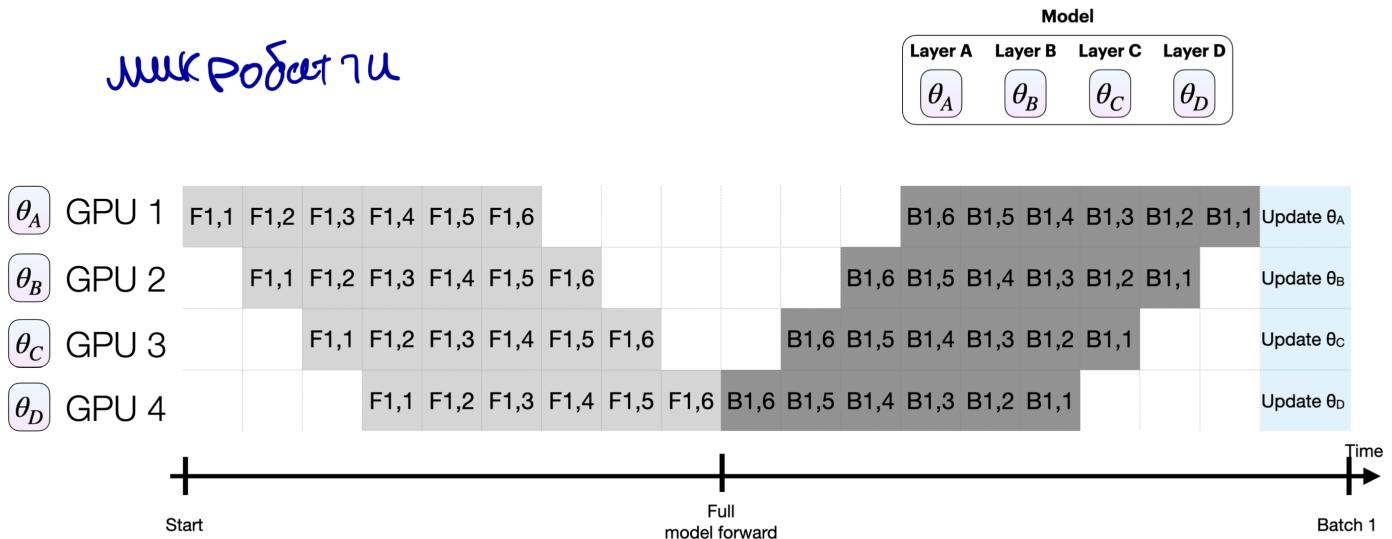


Рис. 5: GPipe

Pipeline model parallelism (PipeDream) ⁹

PipeDream uses asynchronous pipeline parallelism, balancing forward and backward passes across the pipeline stages to maximize utilization and reduce idle time:

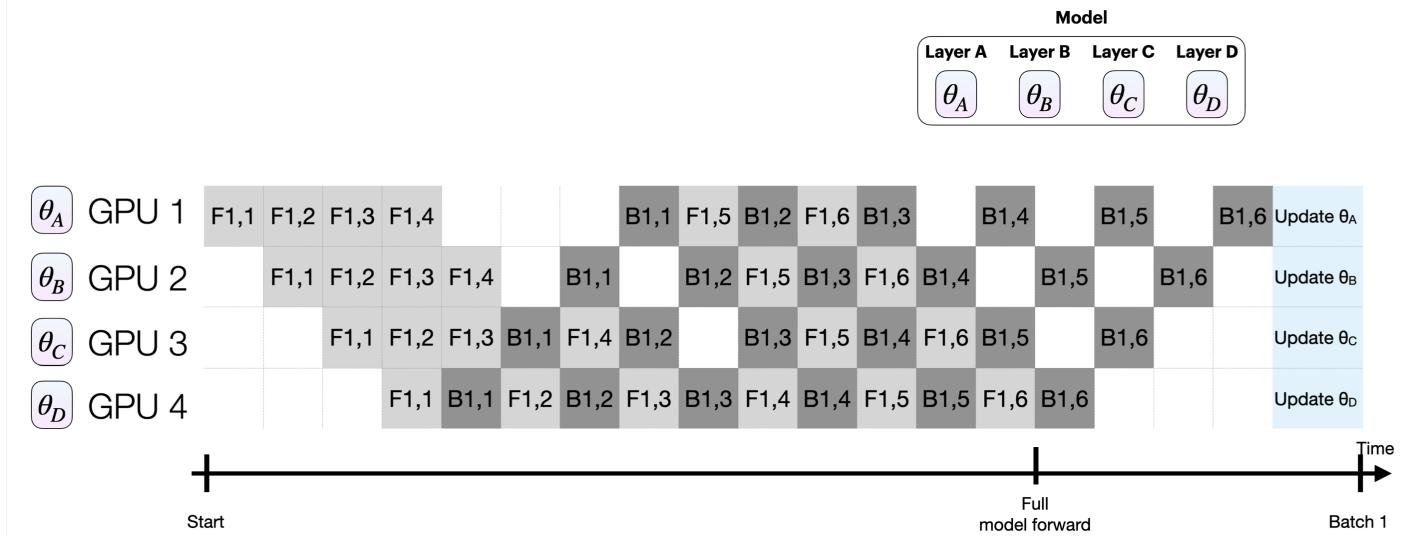
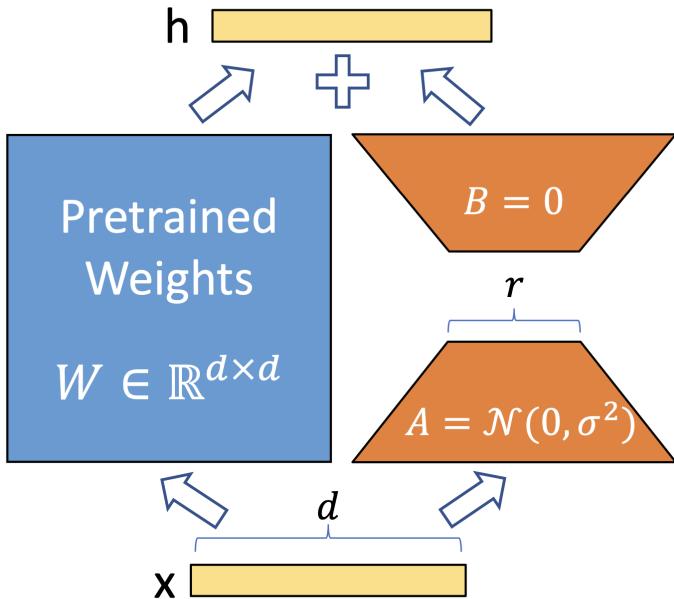


Рис. 6: PipeDream

LoRA¹¹



LoRA reduces the number of parameters by approximating weight matrices with low-rank factorization:

$$W_{\text{new}} = W + \Delta W$$

where $\Delta W = AB^T$, with A and B being low-rank matrices. This reduces computational and memory overhead while maintaining model performance.

- A is initialized as usual, while B is initialized with zeroes in order to start from identity mapping
- r is typically selected between 2 and 64
- Usually applied to attention modules

$$h = W_{\text{new}}x = Wx + \Delta Wx = Wx + AB^Tx$$

¹¹LoRA: Low-Rank Adaptation of Large Language Models

Рис. 7: LoRA

0.153 Метод двойственного градиентного подъема.

Рассматривается задача:

$$f(x) \rightarrow \min_{Ax=b}$$

Двойственная задача:

$$-f^*(-A^T u) - b^T u \rightarrow \max_u$$

где $f^*(y) = \max_x [y^T x - f(x)]$ – сопряженная функция. Определим $g(u) = -f^*(-A^T u) - b^T u$, тогда $\partial g(u) = A\partial f^*(-A^T u) - b$. Перепишем это в виде $\partial g(u) = Ax - b$, где $x \in \arg \min_z [f(z) + u^T Az]$. Тогда определим метод двойственного градиентного подъема:

$$x_k \in \arg \min_x [f(x) + (u_{k-1})^T Ax]$$

$$u_k = u_{k-1} + \alpha_k (Ax_k - b)$$

0.154 Связь константы сильной выпуклости f и гладкости f^* .

Пусть f – замкнутая и выпуклая (функция является замкнутой если её надграфик – замкнутое множество).

Тогда f – сильно выпуклая с константой выпуклости $\mu \longleftrightarrow \nabla f^*$ – липшицев с параметром $\frac{1}{\mu}$.

0.155 Идея dual decomposition.

Рассмотрим

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{при условии, что } Ax = b$$

Здесь $x = (x_1, \dots, x_B) \in \mathbb{R}^n$ делится на B блоков переменных, при этом каждый $x_i \in \mathbb{R}^{n_i}$. Мы также можем соответственно разделить A :

$$A = [A_1 \dots A_B], \quad \text{где } A_i \in \mathbb{R}^{m \times n_i}$$

Наблюдение при вычислении субградиента заключается в том, что минимизация разбивается на B отдельных задач:

$$\begin{aligned} x^{\text{new}} &\in \arg \min_x \left(\sum_{i=1}^B f_i(x_i) + u^T A x \right) \\ \Rightarrow x_i^{\text{new}} &\in \arg \min_{x_i} (f_i(x_i) + u^T A_i x_i), \quad i = 1, \dots, B \end{aligned}$$

$$x_i^k \in \arg \min_{x_i} (f_i(x_i) + (u^{k-1})^T A_i x_i), \quad i = 1, \dots, B$$

$$u^k = u^{k-1} + \alpha_k \left(\sum_{i=1}^B A_i x_i^k - b \right)$$

Эти шаги можно представить следующим образом:

- **Broadcast:** Отправляем u каждому из B процессоров, каждый из которых оптимизирует в параллели, чтобы найти x_i .
- **Gather:** Собираем $A_i x_i$ с каждого процессора, обновляем глобальную двойственную переменную u .

0.156 Метод двойственного градиентного подъема для линейных ограничений-неравенств.

Рассмотрим задачу:

$$\min_x f(x) \quad \text{при условии } Ax = b$$

Ее двойственная задача:

$$\max_u -f^*(-A^T u) - b^T u$$

где f^* является сопряженной функцией для f . Определяя $g(u) = -f^*(-A^T u) - b^T u$, заметим, что:

$$\partial g(u) = A \partial f^*(-A^T u) - b$$

Таким образом, используя наши знания о сопряжённых функциях:

$$\partial g(u) = Ax - b, \quad \text{где } x \in \arg \min_z [f(z) + u^T A z]$$

Метод двойного подъема для максимизации двойственной функции:

$$\boxed{\begin{aligned} x_k &\in \arg \min_x [f(x) + (u_{k-1})^T A x] \\ u_k &= u_{k-1} + \alpha_k (Ax_k - b) \end{aligned}}$$

- Размеры шагов α_k , $k = 1, 2, 3, \dots$, выбираются стандартными способами.
- Проксимальные градиенты и ускорение могут быть применены так, как это обычно делается.

0.157 Метод модифицированной функции Лагранжа.

Недостаток двойного подъема: Сходимость требует сильных условий. Метод модифицированной функции Лагранжа преобразует исходную задачу:

$$\begin{aligned} \min_x f(x) + \frac{\rho}{2} \|Ax - b\|^2 \\ \text{с.т. } Ax = b \end{aligned}$$

где $\rho > 0$ является параметром. Эта формулировка явно эквивалентна исходной задаче. Проблема является сильно выпуклой, если матрица A имеет полный столбцовый ранг.

Двойной градиентный подъем: Итеративные обновления задаются:

$$\begin{aligned} x_k &= \arg \min_x \left[f(x) + (u_{k-1})^T Ax + \frac{\rho}{2} \|Ax - b\|^2 \right] \\ u_k &= u_{k-1} + \rho(Ax_k - b) \end{aligned}$$

0.158 Метод ADMM.

Метод чередующихся направлений множителей или ADMM направлен на объединение лучших методов. Рассмотрим следующую задачу оптимизации:

Минимизировать функцию:

$$\min_{x,z} f(x) + g(z) \quad \text{при условии } Ax + Bz = c$$

Мы дополняем целевую функцию, включив в неё штрафной член за нарушение ограничения:

$$\min_{x,z} f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c\|^2 \quad \text{при условии } Ax + Bz = c$$

где $\rho > 0$ является параметром. Дополненный Лагранжиан для этой задачи определяется как:

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|^2$$

ADMM повторяет следующие шаги для $k = 1, 2, 3, \dots$:

1. **Обновление x :**

$$x_k = \arg \min_x L_\rho(x, z_{k-1}, u_{k-1})$$

2. **Обновление z :**

$$z_k = \arg \min_z L_\rho(x_k, z, u_{k-1})$$

3. **Обновление u :**

$$u_k = u_{k-1} + \rho(Ax_k + Bz_k - c)$$

Замечание: Обычный метод множителей заменил бы первые два шага совместной минимизацией:

$$(x^{(k)}, z^{(k)}) = \arg \min_{x,z} L_\rho(x, z, u^{(k-1)})$$

0.159 Формулировка задачи линейных наименьших квадратов с ℓ_1 регуляризацией в форме ADMM.

Задача линейных наименьших квадратов с ℓ_1 регуляризацией или LASSO (Least Absolute Shrinkage and Selection Operator) может быть формулирована как:

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \quad (2)$$

где $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ и $\lambda > 0$ — это параметр регуляризации.

Для использования ADMM, мы перепишем задачу в эквивалентной форме с разделением переменных:

$$\begin{aligned} \min_{x,z} \quad & \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|z\|_1 \\ \text{при условии} \quad & x = z \end{aligned} \quad (3)$$

Это позволяет применить ADMM для решения этой задачи. Преобразованная задача соответствует следующему дополненному Лагранжиану:

$$L_\rho(x, z, u) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|z\|_1 + u^T(x - z) + \frac{\rho}{2} \|x - z\|_2^2 \quad (4)$$

Здесь u — это двойственные переменные для ограничения $x = z$, а $\rho > 0$ — параметр пенализации.

ADMM повторяет следующие шаги:

1. **Обновление x :**

$$x_{k+1} = \arg \min_x \left(\frac{1}{2} \|Ax - b\|_2^2 + u_k^T(x - z_k) + \frac{\rho}{2} \|x - z_k\|_2^2 \right) \quad (5)$$

2. **Обновление z :**

$$z_{k+1} = \arg \min_z \left(\lambda \|z\|_1 + u_k^T(x_{k+1} - z) + \frac{\rho}{2} \|x_{k+1} - z\|_2^2 \right) \quad (6)$$

Этот шаг представляет собой задачу проксимального оператора для функции ℓ_1 , что эквивалентно soft-thresholding:

$$z_{k+1} = S_{\frac{\lambda}{\rho}} \left(x_{k+1} + \frac{u_k}{\rho} \right) \quad (7)$$

где $S_\kappa(\cdot)$ — оператор мягкого порогирования:

$$S_\kappa(v_i) = \operatorname{sgn}(v_i) \max(|v_i| - \kappa, 0) \quad (8)$$

3. **Обновление u :**

$$u_{k+1} = u_k + \rho(x_{k+1} - z_{k+1}) \quad (9)$$

0.160 Формулировка задачи поиска точки на пересечении двух выпуклых множеств в форме ADMM.

Формулировка задачи поиска точки на пересечении двух выпуклых множеств в форме ADMM включает в себя нахождение точки, которая принадлежит обоим множествам. Пусть имеются два выпуклых множества C_1 и C_2 . Задачу можно интерпретировать как:

Найти x такое, что $x \in C_1$ и $x \in C_2$.

Чтобы применить метод ADMM, мы можем разделить переменные и ввести эквивалентную задачу:

Найти (x, z) такое, что $x \in C_1$, $z \in C_2$ и $x = z$.

Эта задача формулируется как следующая выпуклая оптимизационная задача:

$$\min_{x,z} 0$$

при условии $x \in C_1$, $z \in C_2$, $x = z$.

Для применения ADMM рассматриваем дополненный Лагранжиан:

$$L_\rho(x, z, u) = \delta_{C_1}(x) + \delta_{C_2}(z) + u^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2,$$

где $\delta_{C_1}(x)$ и $\delta_{C_2}(z)$ — индикаторные функции множеств C_1 и C_2 соответственно, которые равны 0, если переменная принадлежит множеству, и ∞ в противном случае, а u — двойственные переменные для ограничения $x = z$, $\rho > 0$ — параметр пенализации.

Полный метод ADMM для поиска точки на пересечении двух выпуклых множеств формулируется как:

$$\begin{aligned} x_{k+1} &= \text{Proj}_{C_1} \left(z_k - \frac{u_k}{\rho} \right), \\ z_{k+1} &= \text{Proj}_{C_2} \left(x_{k+1} + \frac{u_k}{\rho} \right), \\ u_{k+1} &= u_k + \rho(x_{k+1} - z_{k+1}). \end{aligned}$$

1 Критерий положительной определенности матрицы через знаки собственных значений матрицы

Theorem: $A \succcurlyeq (\succ)0 \iff$ все собственные значения матрицы $A \geq (>)0$

→ Пусть некоторые собственные значения λ отрицательны, и x - соответствующий ему собственный вектор.

Тогда:

$$Ax = \lambda x, x^T Ax \geq 0 \rightarrow x^T Ax = \lambda x^T x, x^T x \geq 0 \rightarrow \lambda \geq 0 \text{ - противоречие}$$

← Помним, что положительная определённость задаётся для симметричных матриц. Для симметричной матрицы можем выбрать собственные векторы v_i , образующие ортогональный базис ($i \neq j : v_i^T v_j = 0$ - выкидываем часть слагаемых из суммы в доказательстве). Тогда для $x \in \mathbb{R}^n$

$$x^T Ax = (\alpha_1 v_1 + \dots + \alpha_n v_n)^T A (\alpha_1 v_1 + \dots + \alpha_n v_n) = \sum \alpha_i^2 v_i^T A v_i = \sum \alpha_i^2 v_i^T \lambda v_i$$

Так как $\lambda_i \geq 0$, то и вся сумма неотрицательна.

2 Автоматическое дифференцирование. Вычислительный граф. Forward/Backward mode (в этом вопросе нет доказательств, но необходимо подробно описать алгоритмы).

2.1 Chain rule

Пусть $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Тогда j -ый выход функции f можно представить в виде:

$$\frac{\partial f_j(x_1(t), \dots, x_n(t))}{\partial t} = \sum_{i=1}^n \frac{\partial f_j}{\partial x_i} \frac{\partial x_i}{\partial t} = \sum_{i=1}^n J_{ji} \frac{\partial x_i}{\partial t},$$

где матрица $J \in \mathbb{R}^{m \times n}$ является якобианом функции f . Следовательно, это можно записать в векторной форме как:

$$\frac{\partial f}{\partial t} = J^T \frac{\partial x}{\partial t}$$

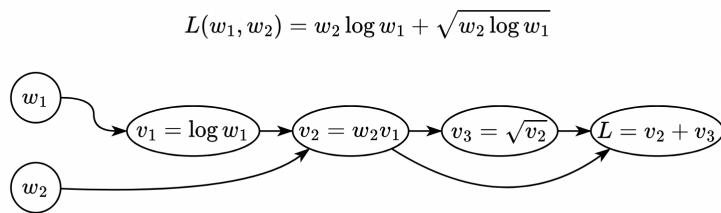
или, эквивалентно,

$$\left(\frac{\partial f}{\partial t} \right)^T = \left(\frac{\partial x}{\partial t} \right)^T J.$$

2.2 Вычислительный граф

Для функции $L(w_1, w_2, \dots, w_d)$ вычислительный граф — ориентированный ациклический граф, с двумя типами вершин: w_1, w_2, \dots, w_d и f_1, f_2, \dots , где w_1, w_2, \dots, w_d обозначают входные переменные, а v_1, v_2, \dots промежуточные функции. В w_1, w_2, \dots, w_d нет входящих ребер, а входящие ребра в v_1, v_2, \dots обозначают значения подающиеся на вход функциям. Также есть одна единственная вершина соответствующая итоговому значению L , из которой не выходит ребер.

Это описание характерной для обучения нейросети функции, так может быть множество вершин стоков L_1, L_2, \dots



Для вычисления $\frac{\partial L}{\partial w_k}$ для всех k есть две конкурирующие процедуры.

2.3 Forward mode

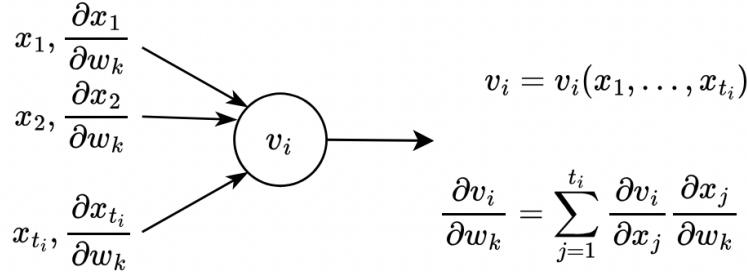
Для каждой w_1, w_2, \dots, w_d делается следующее: для всех i вычисляются $\dot{v}_i = \frac{\partial v_i}{\partial w_k}$. Для этого делается:

- Вычисляется v_i как функцию от её родителей (входов) x_1, \dots, x_{t_i} :

$$v_i = v_i(x_1, \dots, x_{t_i})$$

- Вычисляется производная \dot{v}_i с использованием chain rule:

$$\dot{v}_i = \sum_{j=1}^{t_i} \frac{\partial v_i}{\partial x_j} \frac{\partial x_j}{\partial w_k}$$



Заметим, что для вычисления $\frac{\partial L}{\partial w_k}$ требуется $O(T)$ операций умножения и сложения градиентов, где T — число элементов в вычислительном графе, итого $O(dT)$

2.4 Backward mode

2.4.1 Forward pass

- Вычисляется v_i как функцию от её родителей (входов) x_1, \dots, x_{t_i} :

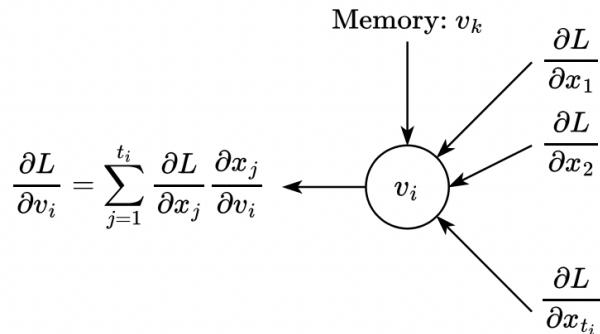
$$v_i = v_i(x_1, \dots, x_{t_i})$$

2.4.2 Backward pass

Обрабатывая вершины в обратном порядке топологического упорядочения:

- Вычислите производную \dot{v}_i используя правило обратной цепочки и информацию от всех его детей (выходов) (x_1, \dots, x_{t_i}) , используются предподсчитанные значения на forward pass:

$$\dot{v}_i = \frac{\partial L}{\partial v_i} = \sum_{j=1}^{t_i} \frac{\partial L}{\partial x_j} \frac{\partial x_j}{\partial v_i}$$



Backward mode быстрее для единственного выхода L . Работает аналогично за линейное время.

3 Метод дихотомии и золотого сечения для унимодальных функций.

Скорость сходимости

Метод дихотомии

Решаем следующую задачу:

$$f(x) \rightarrow \min_{x \in [a,b]}$$

где $f(x)$ – унимодальная функция.

Мы хотим на каждом шаге вдвое сокращать область, в которой ищем минимум. Для этого будем пользоваться основным свойством унимодальных функций:

$$\forall a \leq x_1 < x_2 \leq b :$$

$$f(x_1) \leq f(x_2) \Rightarrow x_* \in [a, x_2]$$

$$f(x_1) \geq f(x_2) \Rightarrow x_* \in [x_1, b]$$

где x_* – точка, в которой достигается минимум

Алгоритм:

```
def binary_search(f, a, b, epsilon):
    c = (a + b) / 2
    while abs(b - a) > epsilon:
        y = (a + c) / 2.0
        if f(y) <= f(c):
            b = c
            c = y
        else:
            z = (b + c) / 2.0
            if f(c) <= f(z):
                a = y
                b = z
            else:
                a = c
                c = z
    return c
```

Можно заметить, что на каждой итерации требуется не более 2-х вычислений значения функции

Сходимость метода дихотомии

Длина отрезка на $k + 1$ итерации:

$$\Delta_{k+1} = b_{k+1} - a_{k+1} = \frac{1}{2^k}(b - a)$$

Если будем выбирать середину отрезка как выход $k + 1$ итерации:

$$|x_{k+1} - x_*| \leq \frac{\Delta_{k+1}}{2}$$

Подставим полученное ранее выражение для длины отрезка:

$$|x_{k+1} - x_*| \leq \frac{1}{2^{k+1}}(b - a)$$

$$|x_{k+1} - x_*| \leq (0.5)^{k+1}(b - a)$$

Получили выражение для сходимости по итерациям. Отсюда также можно выразить необходимое количество итераций для достижения точности ε :

$$K = \left\lceil \log_2 \frac{b - a}{\varepsilon} - 1 \right\rceil$$

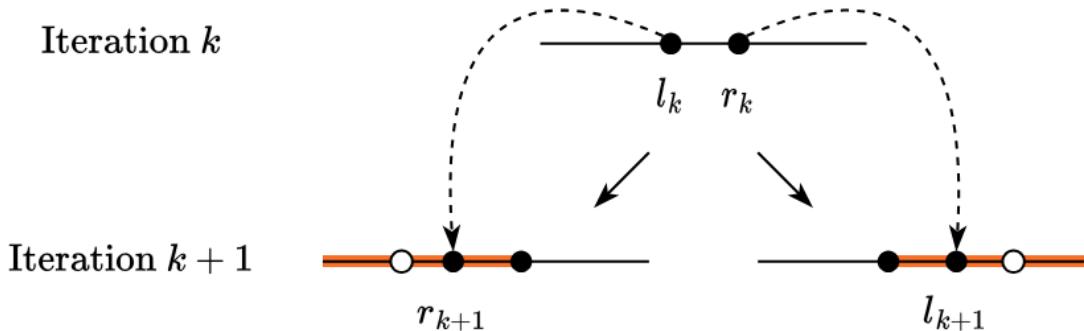
Теперь получим выражение для сходимости по количеству вычислений значения функции. Знаем, что на каждой итерации вычисляем значение не более 2-х раз, значит количество вычислений значения функции возьмём $N = 2k$:

$$|x_{k+1} - x_*| \leq (0.5)^{\frac{N}{2}+1}(b - a)$$

$$|x_{k+1} - x_*| \leq (0.707)^N \frac{b - a}{2}$$

Метод золотого сечения

Идея такая же, как и в методе дихотомии, но хотим уменьшить количество вычислений значения функции. Для этого будем вычислять значения в точках золотого сечения. Так на каждой итерации нам нужно будет вычислять значение только в одной точке, так как для нового отрезка в одной из точек золотого сечения значение будет уже посчитано:



Алгоритм:

```
def golden_search(f, a, b, epsilon):
    tau = (sqrt(5) + 1) / 2
    y = a + (b - a) / tau**2
    z = a + (b - a) / tau
    while b - a > epsilon:
        if f(y) <= f(z):
            b = z
            z = y
            y = a + (b - a) / tau**2
        else:
            a = y
            y = z
            z = a + (b - a) / tau
    return (a + b) / 2
```

Сходимость метода золотого сечения

На каждой итерации длина отрезка будет уменьшаться в $\tau = \frac{\sqrt{5}+1}{2}$ раз. Тогда оценка сходимости (и по итерациям и по вычислениям значений функции):

$$|x_{k+1} - x_*| \leq \frac{b_{k+1} - a_{k+1}}{2} = \left(\frac{1}{\tau}\right)^{N-1} \frac{b - a}{2} \approx 0.618^k \frac{b - a}{2}$$

Получили сходимость по итерациям хуже чем у дихотомии, так как отрезки уменьшаются слабее на каждой итерации. Но по количеству вычислений значения функции, сходимость у метода золотого сечения быстрее.

4 Базовые операции, сохраняющие выпуклость множеств: пересечение бесконечного числа множеств, линейная комбинация множеств, образ афинного отображения.

Пересечение бесконечного числа множеств

Пересечение любого (!) количества выпуклых множеств – выпуклое множество.

Если итоговое пересечение пустое или содержит одну точку, то свойство выпуклости выполняется по определению. Иначе возьмем 2 точки и отрезок между ними. Эти точки должны лежать во всех пересекаемых множествах. Так все пересекаемые множества выпуклы, отрезок между этими двумя точками лежит во всех множествах. А значит отрезок лежит и в их пересечении.

Линейная комбинация множеств

Линейная комбинация выпуклых множеств выпукла.

Пусть есть 2 выпуклых множества S_x, S_y , рассмотрим их линейную комбинацию

$$S = \{s \mid s = c_1x + c_2y, x \in S_x, y \in S_y, c_1, c_2 \in \mathbb{R}\}$$

Возьмем две точки из S : $s_1 = c_1x_1 + c_2y_1, s_2 = c_1x_2 + c_2y_2$ и докажем, что отрезок между ними $\theta s_1 + (1 - \theta)s_2, \theta \in [0, 1]$ также принадлежит S

$$\begin{aligned} & \theta s_1 + (1 - \theta)s_2 \\ & \theta(c_1x_1 + c_2y_1) + (1 - \theta)(c_1x_2 + c_2y_2) \\ & c_1(\theta x_1 + (1 - \theta)x_2) + c_2(\theta y_1 + (1 - \theta)y_2) \\ & c_1x + c_2y \in S \end{aligned}$$

Образ афинного отображения

Образ выпуклого множества после применения афинного отображения – выпуклое множество.

$$S \subseteq \mathbb{R}^n \text{ выпукло} \rightarrow f(S) = \{f(x) \mid x \in S\} \text{ выпукло} \quad (f(x) = \mathbf{A}x + \mathbf{b})$$

Доказательство

При $\theta \in [0, 1]; x, y \in S, S$ – выпуклое. Тогда и $\theta x + (1 - \theta)y \in S$. В то же время $f(\theta x + (1 - \theta)y) = \theta Ax + \theta b + (1 - \theta)Ay + (1 - \theta)b = \theta Ax + (1 - \theta)Ay + b = \theta f(x) + (1 - \theta)f(y)$. В итоге мы доказали, что образ $f(S)$ – тоже выпуклый, так как $\forall \theta \in [0, 1], x, y \in S$ выполняется $\theta f(x) + (1 - \theta)f(y) \in f(S)$

Примеры афинных функций: растяжение, сжатие, проекция, транспонирование, множество решений линейного матричного неравенства $\{x \mid x_1A_1 + \dots + x_mA_m \preceq B\}$. Здесь $A_i, B \in \mathbf{S}^p$ – симметричные матрицы $p \times p$.

Заметим также, что прообраз выпуклого множества при аффинном отображении также является выпуклым.

$$S \subseteq \mathbb{R}^m \text{ convex} \rightarrow f^{-1}(S) = \{x \in \mathbb{R}^n \mid f(x) \in S\} \text{ convex} \quad (f(x) = \mathbf{A}x + \mathbf{b})$$

5 Неравенство Йенсена для выпуклой функции и выпуклой комбинации точек

Вероятностный симплекс:

$\lambda = [\lambda_1, \dots, \lambda_n] \in \Delta_n$, должны выполняться условия:

1. $\forall i \lambda_i \geq 0$

2. $\sum_{i=1}^n \lambda_i = 1$

Неравенство Йенсена для $n = 1$ (считаем известным):

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Неравенство Йенсена:

Пусть $f(x)$ – выпуклая функция, определённая на выпуклом множестве $S \subseteq \mathbb{R}^n$. Тогда для точек $x_1, \dots, x_m \in S$ выполнено неравенство:

$$f\left(\sum_{i=1}^m \lambda_i x_i\right) \leq \sum_{i=1}^m \lambda_i f(x_i)$$

$$\lambda = [\lambda_1, \dots, \lambda_m] \in \Delta_m.$$

Доказательство:

1. Заметим, что $\sum_{i=1}^m \lambda_i x_i$ является выпуклой комбинацией элементов S и лежит в S .

2. Доказательство по индукции. Для $m = 1$ очевидно, для $m = 2$ следует из определения выпуклой функции.

3. Пусть неравенство верно для $m = 1, \dots, k$, докажем для $m = k + 1$. Пусть $\lambda \in \Delta_{k+1}$, $x = \sum_{i=1}^{k+1} \lambda_i x_i = \lambda_{k+1} x_{k+1} + \sum_{i=1}^k \lambda_i x_i$. При $\lambda_i = 0$ либо 1 выражение сводится к уже рассмотренным случаям, далее полагаем $0 < \lambda_i < 1$:

$$x = \lambda_{k+1} x_{k+1} + (1 - \lambda_{k+1}) \sum_{i=1}^k \frac{\lambda_i}{1 - \lambda_{k+1}} x_i = \lambda_{k+1} x_{k+1} + (1 - \lambda_{k+1}) \hat{x}$$

где $\hat{x} = \sum_{i=1}^k \gamma_i x_i$ и $\gamma_i = \frac{\lambda_i}{1 - \lambda_{k+1}} \geq 0$, $1 \geq i \geq k$.

4. Так как $\lambda \in \Delta_{k+1}$, то $\gamma = [\gamma_1, \dots, \gamma_k] \in \Delta_k$. Значит, $\hat{x} \in S$, из выпуклости $f(x)$ и предположения индукции следует:

$$f\left(\sum_{i=1}^{k+1} \lambda_i x_i\right) = f(\lambda_{k+1} x_{k+1} + (1 - \lambda_{k+1}) \hat{x}) \leq \lambda_{k+1} f(x_{k+1}) + (1 - \lambda_{k+1}) f(\hat{x}) \leq \sum_{i=1}^{k+1} \lambda_i f(x_i)$$

6 Выпуклость надграфика как критерий выпуклости функции

Для функции $f(x)$, определенной на $S \subseteq \mathbb{R}^n$, множество:

$$\text{epi } f = \{[x, \mu] \in S \times \mathbb{R} : f(x) \leq \mu\}$$

называется **надграфиком** функции $f(x)$ (здесь $\mu \in \mathbb{R}$, $x \in S$).

Чтобы функция $f(x)$, определенная на выпуклом множестве X , была выпуклой на X , необходимо и достаточно чтобы надграфик f был выпуклым множеством.

Необходимость. Предположим, что $f(x)$ выпукла на X . Возьмем две произвольные точки $[x_1, \mu_1] \in \text{epi } f$ и $[x_2, \mu_2] \in \text{epi } f$. Также возьмем $0 \leq \lambda \leq 1$ и обозначим $x_\lambda = \lambda x_1 + (1 - \lambda)x_2$, $\mu_\lambda = \lambda\mu_1 + (1 - \lambda)\mu_2$. Тогда,

$$\lambda \begin{bmatrix} x_1 \\ \mu_1 \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} x_\lambda \\ \mu_\lambda \end{bmatrix}.$$

Из выпуклости X следует, что $x_\lambda \in X$. Более того, так как $f(x)$ – выпуклая функция, то

$$f(x_\lambda) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \leq \lambda\mu_1 + (1 - \lambda)\mu_2 = \mu_\lambda$$

Из неравенства выше по определению надграфика следует, что $\begin{bmatrix} x_\lambda \\ \mu_\lambda \end{bmatrix} \in \text{epi } f$. Следовательно, надграфик f – выпуклое множество. \square

Достаточность. Предположим, что надграфик f , $\text{epi } f$, выпуклое множество. Тогда, исходя из того что $[x_1, \mu_1] \in \text{epi } f$ и $[x_2, \mu_2] \in \text{epi } f$, получаем

$$\begin{bmatrix} x_\lambda \\ \mu_\lambda \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \mu_1 \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \mu_2 \end{bmatrix} \in \text{epi } f$$

для любого $0 \leq \lambda \leq 1$.

Следовательно, из определения надграфика, подставив значение μ_λ , получаем, что $f(x_\lambda) \leq \mu_\lambda = \lambda\mu_1 + (1 - \lambda)\mu_2$.

$$f(x_\lambda) = f(\lambda x_1 + (1 - \lambda)x_2) \leq \mu_\lambda = \lambda\mu_1 + (1 - \lambda)\mu_2$$

Но это верно для всех $\mu_1 \geq f(x_1)$ и $\mu_2 \geq f(x_2)$, в том числе и при $\mu_1 = f(x_1)$ и $\mu_2 = f(x_2)$. Тогда мы получаем неравенство:

$$f(x_\lambda) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

Так как $x_1 \in X$ и $x_2 \in X$ выбирались произвольно, $f(x)$ – выпуклая функция на X . \square

7 Дифференциальный критерий сильной выпуклости первого порядка

Пусть $f(x)$ — дифференцируемая функция на выпуклом множестве $X \subseteq \mathbb{R}^n$. Тогда $f(x)$ сильно выпукла на X с константой $\mu > 0$ тогда и только тогда, когда

$$f(x) - f(x_0) \geq \langle \nabla f(x_0), x - x_0 \rangle + \frac{\mu}{2} \|x - x_0\|^2$$

для всех $x, x_0 \in X$.

Необходимость. Пусть $0 < \lambda \leq 1$. Согласно определению сильно выпуклой функции,

$$f(\lambda x + (1 - \lambda)x_0) \leq \lambda f(x) + (1 - \lambda)f(x_0) - \frac{\mu}{2}\lambda(1 - \lambda)\|x - x_0\|^2$$

или эквивалентно,

$$\begin{aligned} f(x) - f(x_0) - \frac{\mu}{2}(1 - \lambda)\|x - x_0\|^2 &\geq \frac{1}{\lambda}[f(\lambda x + (1 - \lambda)x_0) - f(x_0)] = \\ &= \frac{1}{\lambda}[f(x_0 + \lambda(x - x_0)) - f(x_0)] = \frac{1}{\lambda}[\lambda\langle \nabla f(x_0), x - x_0 \rangle + o(\lambda)] = \\ &= \langle \nabla f(x_0), x - x_0 \rangle + \frac{o(\lambda)}{\lambda}. \end{aligned}$$

Таким образом, переходя к пределу при $\lambda \rightarrow 0$, мы приходим к первоначальному утверждению. \square

Достаточность. Предположим, что неравенство в теореме выполнено для всех $x, x_0 \in X$. Возьмем $x_0 = \lambda x_1 + (1 - \lambda)x_2$, где $x_1, x_2 \in X$, $0 \leq \lambda \leq 1$. Согласно неравенству из условия теоремы, выполняются следующие неравенства:

$$f(x_1) - f(x_0) \geq \langle \nabla f(x_0), x_1 - x_0 \rangle + \frac{\mu}{2}\|x_1 - x_0\|^2,$$

$$f(x_2) - f(x_0) \geq \langle \nabla f(x_0), x_2 - x_0 \rangle + \frac{\mu}{2}\|x_2 - x_0\|^2.$$

Умножая первое неравенство на λ и второе на $1 - \lambda$ и складывая их, учитывая, что

$$x_1 - x_0 = (1 - \lambda)(x_1 - x_2), \quad x_2 - x_0 = \lambda(x_2 - x_1),$$

и что $\lambda(1 - \lambda)^2 + \lambda^2(1 - \lambda) = \lambda(1 - \lambda)$, получаем:

$$\begin{aligned} \lambda f(x_1) + (1 - \lambda)f(x_2) - f(x_0) - \frac{\mu}{2}\lambda(1 - \lambda)\|x_1 - x_2\|^2 &\geq \\ &\geq \langle \nabla f(x_0), \lambda x_1 + (1 - \lambda)x_2 - x_0 \rangle = 0. \end{aligned}$$

Таким образом, неравенство из определения сильно выпуклой функции выполнено. Важно отметить, что при $\mu = 0$ получаем случай выпуклости и соответствующий дифференциальный критерий. \square

8 Дифференциальный критерий сильной выпуклости второго порядка

Пусть $X \subseteq \mathbb{R}^n$ — выпуклое множество с непустой внутренностью. Пусть также $f(x)$ — дважды непрерывно дифференцируемая функция на X . Тогда $f(x)$ сильно выпукла на X с константой $\mu > 0$ тогда и только тогда, когда

$$\langle y, \nabla^2 f(x)y \rangle \geq \mu \|y\|^2$$

для всех $x \in X$ и $y \in \mathbb{R}^n$.

Другая форма записи:

$$\nabla^2 f(x) \succ \mu I$$

Целевое неравенство тривиально, когда $y = 0_n$, поэтому предположим, что $y \neq 0_n$.

Необходимость. Пусть x является внутренней точкой X . Тогда $x + \alpha y \in X$ для всех $y \in \mathbb{R}^n$ и достаточно малых α . Поскольку $f(x)$ дважды дифференцируема,

$$f(x + \alpha y) = f(x) + \alpha \langle \nabla f(x), y \rangle + \frac{\alpha^2}{2} \langle y, \nabla^2 f(x)y \rangle + o(\alpha^2).$$

Основываясь на критерии первого порядка сильной выпуклости, имеем

$$\frac{\alpha^2}{2} \langle y, \nabla^2 f(x)y \rangle + o(\alpha^2) = f(x + \alpha y) - f(x) - \alpha \langle \nabla f(x), y \rangle \geq \frac{\mu}{2} \alpha^2 \|y\|^2.$$

Это неравенство сводится к целевому неравенству после деления обеих частей на α^2 и перехода к пределу при $\alpha \rightarrow 0$.

Если $x \in X$, но $x \notin \text{int } X$, рассмотрим последовательность $\{x_k\}$ такую, что $x_k \in \text{int } X$ и $x_k \rightarrow x$ при $k \rightarrow \infty$. Тогда мы приходим к целевому неравенству после перехода к пределу. \square

Достаточность. Формула Тейлора с остаточным членом Лагранжа второго порядка $\forall x, y : x, x+y \in X$ найдется α такая, что:

$$f(x+y) = f(x) + \langle \nabla f(x), y \rangle + \frac{1}{2} \langle y, \nabla^2 f(x+\alpha y)y \rangle$$

где $0 < \alpha < 1$.

Используя формулу Тейлора с остаточным членом Лагранжа и неравенство из условия, получаем для $x+y \in X$:

$$f(x+y) - f(x) - \langle \nabla f(x), y \rangle = \frac{1}{2} \langle y, \nabla^2 f(x+\alpha y)y \rangle \geq \frac{\mu}{2} \|y\|^2,$$

где $0 \leq \alpha \leq 1$. Следовательно,

$$f(x+y) - f(x) \geq \langle \nabla f(x), y \rangle + \frac{\mu}{2} \|y\|^2.$$

Таким образом, по критерию первого порядка сильной выпуклости, функция $f(x)$ является сильно выпуклой с константой μ . Важно отметить, что $\mu = 0$ соответствует случаю выпуклости и соответствующему дифференциальному критерию. \square

9 Необходимые условия безусловного экстремума.

Если в x^* достигается локальный минимум и f непрерывно дифференцируема в открытой окрестности, то

$$\nabla f(x^*) = 0$$

Доказательство

Предположим обратное. Пусть $\nabla f(x^*) \neq 0$. Рассмотрим вектор $p = -\nabla f(x^*)$ и заметим, что

$$p^T \nabla f(x^*) = -\|\nabla f(x^*)\|^2 < 0$$

Так как ∇f непрерывна в окрестности x^* , то существует скаляр $T > 0$ такой, что

$$p^T \nabla f(x^* + tp) < 0, \text{ для любого } t \in [0, T]$$

Для любого $\bar{t} \in (0, T]$, мы можем воспользоваться теоремой Тейлора:

$$f(x^* + \bar{t}p) = f(x^*) + \bar{t}p^T \nabla f(x^* + tp), \text{ для некоторого } t \in (0, \bar{t})$$

Следовательно, $f(x^* + \bar{t}p) < f(x^*)$ для любого $\bar{t} \in (0, T]$. Мы нашли направление, идя вдоль которого из x^* функция f убывает. Тогда x^* – не точка локального минимума. Получили противоречие.

10 Достаточные условия безусловного экстремума.

Пусть $\nabla^2 f$ непрерывна в открытой окрестности x^* и

$$\nabla f(x^*) = 0 \quad \nabla^2 f(x^*) \succ 0.$$

Тогда x^* – точка локального минимума f .

Доказательство

Так как гессиан непрерывен и положительно определен в x^* , то мы можем выбрать радиус $r > 0$ такой, что $\nabla^2 f(x)$ остается положительно определенной для всех x в открытом шаре $B = \{z \mid \|z - x^*\| < r\}$. Взяв любой ненулевой вектор p , для которого выполняется $\|p\| < r$, мы получаем $x^* + p \in B$, а также по формуле Тейлора:

$$\begin{aligned} f(x^* + p) &= f(x^*) + p^T \nabla f(x^*) + \frac{1}{2} p^T \nabla^2 f(z)p \\ &= f(x^*) + \frac{1}{2} p^T \nabla^2 f(z)p \end{aligned}$$

где $z = x^* + tp$ для некоторого $t \in (0,1)$. Так как $z \in B$, мы получаем $p^T \nabla^2 f(z)p > 0$, и следовательно $f(x^* + p) > f(x^*)$. Таким образом x^* – точка локального минимума.

11 Формулировка симплекс метода для задачи линейного программирования в стандартной форме. Теорема о проверке оптимальности решения.

Задача линейного программирования в стандартном виде:

Пусть $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, тогда задача формулируется так:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x_i \geq 0, \quad i = 1, \dots, n \end{aligned}$$

Идейное описание симплекс метода:

1. Убедится, что точка, в которой мы находимся, является угловой
2. Проверить оптимальность точки
3. Если необходимо, сменить угол (то есть сменить базис)
4. Повторять до схождения

Теорема о проверке оптимальности решения:

Если все элементы λ_B неположительны и базис B достижим, тогда базис B оптimalен.

Здесь λ_B это коэффициенты при разложении c по базису B : $\lambda_B^T A_B = c^T \Rightarrow \lambda_B^T = c^T A_B^{-1}$.

Доказательство:

Предположим противное (что этот базис не оптimalен), пусть $\exists x^* : Ax^* \leq b$ и при этом $c^T x^* < c^T x_B$. Так как для всей матрицы A и вектора b неравенство верно, то и для подматрицы оно верно:

$$A_B x^* \leq b_B$$

Так как все элементы λ_B неположительны, то домножим строки на соответствующие элементы и сложим:

$$\lambda_B^T A_B x^* \geq \lambda_B^T b_B$$

$$c^T x^* \geq \lambda_B^T b_B = \lambda_B^T A_B x_B = c^T x_B$$

Противоречие.

12 Теорема сходимости градиентного спуска для гладких выпуклых функций.

Теорема:

Рассматриваем задачу:

$$f(x) \rightarrow \min_{x \in \mathbb{R}^n}$$

И предполагаем, что f - выпуклая, L -гладкая, $L > 0$. Тогда градиентный спуск с постоянным шагом $0 < \alpha \leq \frac{1}{L}$ будет сходиться сублинейно:

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

Доказательство:

L -гладкость:

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$$

Возьмём $y := x_{k+1}, x = x_k$:

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2$$

Заметим, что $x_{k+1} - x_k = -\alpha \nabla f(x_k)$. Подставим:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), -\alpha \nabla f(x_k) \rangle + \frac{L}{2} \alpha^2 \|\nabla f(x_k)\|^2 \\ f(x_{k+1}) &\leq f(x_k) - \alpha \|\nabla f(x_k)\|^2 + \frac{L}{2} \alpha^2 \|\nabla f(x_k)\|^2 \\ f(x_{k+1}) &\leq f(x_k) + \left(\frac{L}{2} \alpha^2 - \alpha \right) \|\nabla f(x_k)\|^2 \end{aligned} \tag{10}$$

Отсюда получаем оптимальный размер шага:

$$\frac{L}{2} \alpha^2 - \alpha \rightarrow \min_{\alpha}$$

$$\alpha^* = \frac{1}{L}$$

Можно заметить, что $\frac{L}{2} \alpha^2 - \alpha = -\frac{\alpha}{2}(2 - L\alpha) \leq -\frac{\alpha}{2}$ для $\alpha \leq \frac{1}{L}$. Подставим $-\frac{\alpha}{2}$:

$$f(x_{k+1}) \leq f(x_k) - \frac{\alpha}{2} \|\nabla f(x_k)\|^2 \tag{11}$$

Выпуклость:

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

Возьмём $y := x^*, x := x_k$:

$$f(x^*) \geq f(x_k) + \langle \nabla f(x_k), x^* - x_k \rangle$$

$$f(x_k) \leq f(x^*) + \langle \nabla f(x_k), x_k - x^* \rangle \tag{12}$$

Подставляем (3) в (2):

$$\begin{aligned} f(x_{k+1}) &\leq f(x^*) + \langle \nabla f(x_k), x_k - x^* \rangle - \frac{\alpha}{2} \|\nabla f(x_k)\|^2 \\ f(x_{k+1}) &\leq f(x^*) + \left\langle \nabla f(x_k), x_k - x^* - \frac{\alpha}{2} \nabla f(x_k) \right\rangle \\ f(x_{k+1}) &\leq f(x^*) + \frac{1}{2\alpha} \left\langle \alpha \nabla f(x_k), 2 \left(x_k - x^* - \frac{\alpha}{2} \nabla f(x_k) \right) \right\rangle \end{aligned}$$

Возьмём $a = x_k - x^*$, $b = x_k - x^* - \alpha \nabla f(x_k)$. Можно заметить, что:

$$\forall x, y \in \mathbb{R}^n : \quad \langle x - y, x + y \rangle = \|x\|^2 - \|y\|^2$$

$$\begin{aligned} a - b &= \alpha \nabla f(x_k) \\ a + b &= 2 \left(x_k - x^* - \frac{\alpha}{2} \nabla f(x_k) \right) \end{aligned}$$

Получаем:

$$f(x_{k+1}) \leq f(x^*) + \frac{1}{2\alpha} (\|x_k - x^*\|^2 - \|x_k - x^* - \alpha \nabla f(x_k)\|^2)$$

Так как $x_{k+1} = x_k - \alpha \nabla f(x_k)$:

$$f(x_{k+1}) - f(x^*) \leq \frac{1}{2\alpha} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2)$$

Продуммируем:

$$\begin{aligned} \sum_{i=0}^{k-1} (f(x_{i+1}) - f(x^*)) &\leq \frac{1}{2\alpha} \sum_{i=0}^{k-1} (\|x_i - x^*\|^2 - \|x_{i+1} - x^*\|^2) \\ \sum_{i=0}^{k-1} (f(x_{i+1}) - f(x^*)) &\leq \frac{1}{2\alpha} (\|x_0 - x^*\|^2 - \|x_k - x^*\|^2) \\ \sum_{i=0}^{k-1} (f(x_{i+1}) - f(x^*)) &\leq \frac{1}{2\alpha} \|x_0 - x^*\|^2 \\ \frac{1}{k} \sum_{i=0}^{k-1} f(x_{i+1}) - f(x^*) &\leq \frac{1}{2\alpha k} \|x_0 - x^*\|^2 \end{aligned}$$

Из (2) знаем, что $f(x_{i+1}) \leq f(x_i)$. Тогда $f(x_k) \leq \frac{1}{k} \sum_{i=1}^{k-1} f(x_i)$. Подставим $f(x_k)$ вместо суммы:

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

Получили сублинейную сходимость для $0 < \alpha \leq \frac{1}{L}$

13 Теорема сходимости градиентного спуска для гладких PL функций.

Требования: f is μ -PL и L -гладкая, $L \geq \mu > 0$.

Тогда решаемая задача $f(x) \rightarrow \min_{x \in \mathbb{R}^d}$ с константным lr $\alpha \in (0, \frac{1}{L}]$ и градиентным спуском $x_{k+1} = x_k - \alpha \nabla f(x_k)$ имеет линейную сходимость, т.е.:

$$f(x_k) - f^* \leq (1 - \alpha\mu)^k (f(x_0) - f^*)$$

Доказательство :

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

$$L\text{-гладкость: } \forall x, y : f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$$

$$y := x_{k+1}, x := x_k \implies f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), -\alpha \nabla f(x_k) \rangle + \frac{L}{2} \|\alpha \nabla f(x_k)\|^2$$

$$f(x_{k+1}) \leq f(x_k) - \alpha \|\nabla f(x_k)\|^2 + \frac{L\alpha^2}{2} \|\nabla f(x_k)\|^2$$

$$(L\alpha \leq 1)$$

$$f(x_{k+1}) \leq f(x_k) - \alpha \|\nabla f(x_k)\|^2 + \frac{\alpha}{2} \|\nabla f(x_k)\|^2 = f(x_k) - \frac{\alpha}{2} \|\nabla f(x_k)\|^2$$

$$\text{Применим условие PL: } \|\nabla f(x_k)\|^2 \geq 2\mu(f(x_k) - f^*)$$

$$f(x_{k+1}) - f^* \leq f(x_k) - f^* - \alpha\mu(f(x_k) - f^*) = (1 - \alpha\mu)(f(x_k) - f^*) = \dots = (1 - \alpha\mu)^{k+1}(f(x_0) - f^*)$$

То есть $f(x_k) - f^* \leq (1 - \alpha\mu)^k (f(x_0) - f^*)$ и характер сходимости – линейный.

14 Теорема сходимости градиентного спуска для сильно выпуклых квадратичных функций. Оптимальные гиперпараметры.

Теорема:

$$f \rightarrow \min_{x \in \mathbb{R}^n}$$

$$f(x) = x^\top A x - b^\top x + c, \quad A \in S_{++}^n$$

Тогда градиентный спуск с шагом $\alpha_{opt} = \frac{2}{\mu+L}$ сходится линейно с показателем $\frac{L-\mu}{L+\mu}$:

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

Доказательство:

Градиент функции $f(x)$:

$$\nabla f(x) = Ax - b$$

Тогда шаг градиентного спуска имеет вид:

$$x_{k+1} = x_k - \alpha(Ax_k - b)$$

Найдём оптимальный размер шага. Для начала преобразуем шаг:

$$x_{k+1} = (I - \alpha A)x_k + \alpha b$$

Выразим b через x_* :

$$\nabla f(x_*) = 0$$

$$Ax_* - b = 0$$

$$b = Ax_*$$

Подставим и преобразуем:

$$x_{k+1} = (I - \alpha A)x_k + \alpha Ax_*$$

$$x_{k+1} - x_* = (I - \alpha A)x_k + \alpha Ax_* - x_*$$

$$x_{k+1} - x_* = (I - \alpha A)(x_k - x_*)$$

Воспользуемся спектральным разложением матрицы $A = Q\Lambda Q^\top$, где $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, $Q = [q_1, \dots, q_n]$, λ_i, q_i - i -ые собственное значение и собственный вектор. Знаем, что матрица Q - ортогональная ($QQ^\top = Q^\top Q = I$).

$$x_{k+1} - x_* = (I - \alpha Q\Lambda Q^\top)(x_k - x_*)$$

Домножим обе части равенства на Q^\top слева:

$$Q^\top(x_{k+1} - x_*) = (Q^\top - \alpha Q^\top Q \Lambda Q^\top)(x_k - x_*)$$

$$Q^\top(x_{k+1} - x_*) = (Q^\top - \alpha \Lambda Q^\top)(x_k - x_*)$$

$$Q^\top(x_{k+1} - x_*) = (I - \alpha \Lambda) Q^\top(x_k - x_*)$$

Сделаем замену $\tilde{x} = Q^\top(x - x_*)$:

$$\tilde{x}_{k+1} = (I - \alpha \Lambda) \tilde{x}_k$$

Получаем для i -й координаты:

$$\tilde{x}_i^{k+1} = (1 - \alpha \lambda_i) \tilde{x}_i^k$$

Обозначим $\mu = \lambda_{\min} > 0$, $L = \lambda_{\max} > 0$, $\rho(\alpha) = \max_i |1 - \alpha \lambda_{(i)}|$. Условие сходимости: $\rho(\alpha) < 1$. Распишем крайние случаи для $\rho(\alpha)$:

$$\begin{array}{ll} |1 - \alpha \mu| < 1 & |1 - \alpha L| < 1 \\ -1 < 1 - \alpha \mu < 1 & -1 < 1 - \alpha L < 1 \\ 0 < \alpha < \frac{2}{\mu} & 0 < \alpha < \frac{2}{L} \end{array}$$

Получили, что для сходимости необходимо, чтобы выполнялось $\alpha < \frac{2}{L}$. Теперь вычислим оптимальный шаг.

Хотим минимизировать $\rho(\alpha)$:

$$\rho^* = \min_{\alpha} \rho(\alpha) = \min_{\alpha} \max_i |1 - \alpha \lambda_{(i)}| = \min_{\alpha} \max \{|1 - \alpha \mu|, |1 - \alpha L|\}$$

Отсюда получаем, что при оптимальном выборе шага должно выполняться $|1 - \alpha^* \mu| = |1 - \alpha^* L|$. Вычислим α^* :

$$\begin{aligned} 1 - \alpha^* \mu &= \alpha^* L - 1 \\ \alpha^* = \frac{2}{\mu + L} &\Rightarrow \rho^* = \frac{L - \mu}{L + \mu} \end{aligned}$$

Получаем выражение для сходимости:

$$\begin{aligned} \tilde{x}_i^{k+1} &\leq \rho^* \tilde{x}_i^k \\ \tilde{x}_i^{k+1} &\leq (\rho^*)^{k+1} \tilde{x}_i^0 \\ \|x_{k+1} - x_*\| &\leq \left(\frac{L - \mu}{L + \mu}\right)^k \|x_0 - x_*\| \end{aligned}$$

Получили линейную сходимость с показателем $\frac{L - \mu}{L + \mu}$

- 15 Теорема сходимости субградиентного метода для выпуклых функций. Сходимость метода для разных стратегий выбора шага: постоянный размер шага $\alpha_k = \alpha$; Обратный квадратный корень $\frac{R}{G\sqrt{k}}$; Обратный $\frac{1}{k}$; Размер шага Поляка: $\alpha_k = \frac{f(x^k) - f^*}{\|g_k\|_2^2}$.

16 Теорема о сходимости метода тяжелого шарика для сильно выпуклой квадратичной задачи.

Теорема: Рассматриваем задачу:

$$f \rightarrow \min_{x \in \mathbb{R}^n}$$

$$f(x) = \frac{1}{2} x^\top A x - b^\top x + c, \quad A \in \mathbb{S}_{++}^{n \times n}$$

Не умаляя общности, считаем $c = 0$, т.к. ответ от него не зависит. Метод тяжёлого шарика имеет вид:

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

Тогда скорость сходимости не зависит от шага (при допустимых его значениях). Показатель сходимости $q \sim \sqrt{\beta^*}$, где β^* – оптимальный гиперпараметр.

$$\beta^* = \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^2$$

$$\|x_k - x^*\| \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x_0 - x^*\|$$

Доказательство:

$$\nabla f(x) = Ax - b \Rightarrow Ax^* = b$$

Воспользуемся спектральным разложением матрицы $A = Q\Lambda Q^\top$, где $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, $Q = [q_1, \dots, q_n]$, λ_i, q_i – i -ые собственное значение и собственный вектор. Знаем, что матрица Q – ортогональная ($QQ^\top = Q^\top Q = I$). Сделаем замену:

$$\begin{aligned} \tilde{x} &= Q^\top(x - x^*) \Rightarrow x = Q\tilde{x} + x^* \\ f(\tilde{x}) &= \frac{1}{2} (Q\tilde{x} + x^*)^\top A (Q\tilde{x} + x^*) - b^\top (Q\tilde{x} + x^*) \\ &= \frac{1}{2} \tilde{x}^\top Q^\top A Q \tilde{x} + (x^*)^\top A Q \tilde{x} + \frac{1}{2} (x^*)^\top A x^* - b^\top Q \tilde{x} - b^\top x^* \\ &= \tilde{x}^\top \Lambda \tilde{x} - \frac{1}{2} b^\top x^* \end{aligned}$$

Можем отбросить константный член в последнем выражении, т.к. он не будет влиять на оптимизацию:

$$f(\tilde{x}) = \frac{1}{2} \tilde{x}^\top \Lambda \tilde{x}$$

$$\nabla f(\tilde{x}) = \Lambda \tilde{x}$$

Можем переписать правило обновления:

$$\begin{cases} \tilde{x}_{k+1} = (I + \alpha \Lambda - \beta I) \tilde{x}_k - \beta x_{k-1} \\ \tilde{x}_k = \tilde{x}_k \end{cases} \quad (13)$$

Рассмотрим $\tilde{z}_k = \begin{pmatrix} \tilde{x}_{k+1} \\ \tilde{x}_k \end{pmatrix}$. Тогда правило обновления имеет вид:

$$\tilde{z}_{k+1} = M \tilde{z}_k$$

$$M = \begin{pmatrix} I - \alpha\Lambda + \beta I & -\beta I \\ I & 0 \end{pmatrix} \in \mathbb{R}^{2n \times 2n}$$

Сделаем reshape:

$$\begin{pmatrix} \tilde{x}_{k+1}^{(1)} \\ \tilde{x}_{k+1}^{(2)} \\ \vdots \\ \tilde{x}_{k+1}^{(n)} \\ \tilde{x}_k^{(1)} \\ \tilde{x}_k^{(2)} \\ \vdots \\ \tilde{x}_k^{(n)} \end{pmatrix} \rightarrow \begin{pmatrix} \tilde{x}_{k+1}^{(1)} \\ \tilde{x}_k^{(1)} \\ \tilde{x}_{k+1}^{(2)} \\ \tilde{x}_k^{(2)} \\ \vdots \\ \tilde{x}_{k+1}^{(n)} \\ \tilde{x}_k^{(n)} \end{pmatrix} \Rightarrow M = \begin{pmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_n \end{pmatrix}$$

Для i -й координаты:

$$M_i = \begin{pmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{pmatrix}$$

Метод сходится при $\rho(M) < 1$, где $\rho(M)$ – спектральный радиус. Оптимальные гиперпараметры могут быть получены при оптимизации спектрального радиуса:

$$\alpha^*, \beta^* = \arg \min_{\alpha, \beta} \rho(M) = \arg \min_{\alpha, \beta} \max_i \rho(M_i)$$

Собственные значения M_i имеют вид:

$$\lambda_1^{M_i}, \lambda_2^{M_i} = \frac{1 + \beta - \alpha\lambda_i \pm \sqrt{(1 + \beta - \alpha\lambda_i)^2 - 4\beta}}{2} \quad (14)$$

При оптимизации получаем:

$$\alpha^* = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2} \quad \beta^* = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2$$

При (α^*, β^*) собственные значения получаются комплексно сопряжёнными. Подставляя (α^*, β^*) в 14 получаем:

$$q = |\lambda_1^{M_i}| = \frac{L - \mu}{\left(\sqrt{L} + \sqrt{\mu} \right)^2} = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} = \sqrt{\beta^*}$$

То есть показатель сходимости не зависит от α^* и равен $\sqrt{\beta^*}$. Тогда можно записать:

$$\|\tilde{z}_k - \tilde{z}_*\| \leq \sqrt{\beta^*} \|\tilde{z}_{k-1} - \tilde{z}_*\|$$

$$\|\tilde{z}_k - \tilde{z}_*\| \leq (\sqrt{\beta^*})^k \|\tilde{z}_0 - \tilde{z}_*\|$$

Итого получаем:

$$\|x_k - x^*\| \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x_0 - x^*\|$$

17 Теорема о сходимости метода проекции градиента для выпуклой гладкой функции.

18 Теорема о сходимости метода проекции градиента для сильно выпуклой гладкой функции.

Рассматриваем задачу

$$f(x) \rightarrow \min_{x \in S},$$

где f - μ -сильно выпуклая и L -гладкая. Пусть $S \subseteq \mathbb{R}^n$ замкнутое выпуклое множество. Тогда метод проекции градиента с постоянным шагом $\alpha \leq \frac{1}{L}$:

$$x_{k+1} = \text{proj}_S(x_k - \alpha \nabla f(x_k))$$

сходится со линейно и $\forall T$ выполняется неравенство:

$$f(x_T) - f^* \leq (1 - \alpha\mu)^T (f(x_0) - f^*).$$

Повторяет доказательство теоремы 17, с заменой оператора prox на proj.

19 Теорема о сходимости метода Франк-Вульфа для выпуклой гладкой функции.

20 Доказательство сходимости метода сопряженных градиентов и вывод формулы.

21 Теорема сходимости метода Ньютона для сильно выпуклых функций с липшицевым гессианом.

22 Вывод формул обновления оценок обратного гессиана и гессиана квазиньютоновских методов SR-1, DFP, BFGS.

23 Теорема о сходимостиproxимального градиентного для выпуклой гладкой функции f .

24 Теорема о сходимости проксимального градиентного для сильно выпуклой гладкой функции f .

25 Теорема о сходимости стохастического градиентного спуска в гладком PL-случае.