

Определения и формулировки

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.

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. p -норма вектора)

3. 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: $\|x\|_1 = \sum_{i=0}^n |x_i|$

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

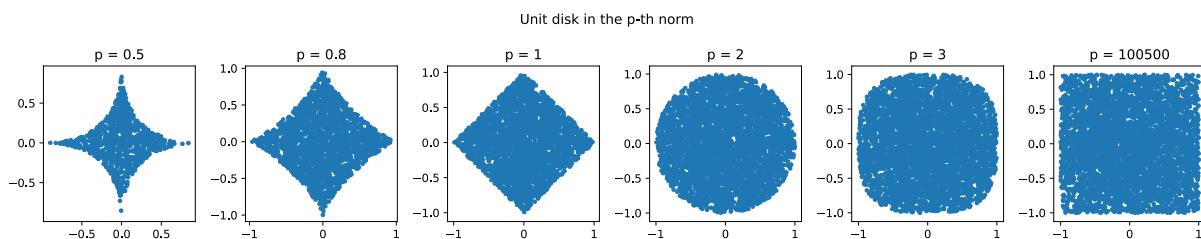


Рисунок 1: Шары в разных нормах

5. Норма Фробениуса для матрицы.



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

6. 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 $A^\top A$.

7. Скалярное произведение двух векторов.



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

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

8. 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: $\langle X, X \rangle = \|X\|_F^2$

9. Собственные значения матрицы. Спектр матрицы.



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

Спектр матрицы – совокупность её собственных значений.

10. Связь спектра матрицы и её определенности.



Матрица положительно (неотрицательно) определена \Leftrightarrow её спектр (все её собственные значения) положителен (неотрицателен).

11. Спектральное разложение матрицы.

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

$$A_{n \times n} = S_{n \times n} \begin{matrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_n \end{matrix} S_{n \times n}^{-1}$$

Рисунок 2: Спектральное разложение матрицы

12. Сингулярное разложение матрицы.

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

$$A = U\Sigma V^T$$

$U \in \mathbb{R}^{m \times r}$, $U^T U = I$, $V \in \mathbb{R}^{n \times r}$, $V^T V = I$, 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 значения.

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

$$A_{m \times n} = U_{m \times m} \begin{matrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_r \end{matrix} \Sigma_{m \times n} V_{n \times n}^*$$

Рисунок 3: Сингулярное разложение матрицы

13. Связь определителя и собственных чисел для квадратной матрицы.

💡 Если у матрицы A собственные значения $\lambda_1, \lambda_2, \dots, \lambda_n$, то её определитель равен:

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

14. Связь следа и собственных чисел для квадратной матрицы.

💡 Если у матрицы A собственные значения $\lambda_1, \lambda_2, \dots, \lambda_n$, то её след равен:

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

15. Линейная сходимость последовательности.

💡 Пусть есть последовательность $\{r_k\}_{k=m}^{\infty} = \{\|x_k - x^*\|_2\}$ в \mathbb{R} , сходящаяся к 0.

Линейная сходимость последовательности $\{r_k\}_{k=m}^{\infty}$ определяется следующим образом: Если существует константа $C > 0$ такая, что:

$$r_k \leq Cq^k, \quad \text{для всех } k \geq m.$$

Если такое q существует, то последовательность называется **линейно сходящейся**. Точная нижняя граница всех q , удовлетворяющих неравенству, называется **скоростью линейной сходимости** последовательности.

16. 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:

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

где $q < 0$ и $0 < C < \infty$.

17. Сверхлинейная сходимость последовательности.

💡 Мы определяем сверхлинейную сходимость как сходимость последовательности, которая быстрее любой линейной сходимости. Иногда рассматривают более специальный класс. Тогда говорят, что сверхлинейная сходимость при $q > 1, C > 0$ определяется следующим образом:

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

18. Квадратичная сходимость последовательности.

💡 Последовательность $\{r_k\}_{k=m}^{\infty}$ называется **сходящейся сверхлинейно**, если она сходится к нулю быстрее любой линейно сходящейся последовательности.

Для $p > 1$, последовательность имеет **сверхлинейную сходимость порядка p** , если существуют $C > 0$ и $0 < q < 1$ такие, что:

$$r_k \leq Cq^{p^k}, \quad \text{для всех } k \geq m.$$

Когда $p = 2$, это называется **квадратичной сходимостью**.

19. Рассмотрим итеративный процесс $x_k \in \mathbb{R}$, сходящийся к решению $x^* \in \mathbb{R}$ квадратично. Как изменяется количество верных значащих цифр в решении после одной итерации метода?

💡 Удваивается.

Пояснение: Пусть $x^* = 1.23456789012345$ (истинное решение), и итерационная последовательность начинается с ошибкой $r_k = 10^{-3}$, соответствующей 3 верным значащим цифрам (1.234).

После первой итерации:

$$r_{k+1} \approx r_k^2 = (10^{-3})^2 = 10^{-6}.$$

Теперь ошибка равна 10^{-6} , и мы имеем 6 верных значащих цифр (1.23456).

После второй итерации:

$$r_{k+2} \approx r_{k+1}^2 = (10^{-6})^2 = 10^{-12}.$$

Теперь ошибка равна 10^{-12} , и мы имеем 12 верных значащих цифр (1.234567890123).

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 $\alpha \geq 0$.)

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

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.

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 $x^* < x_1 < x_2 \leq b$

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.

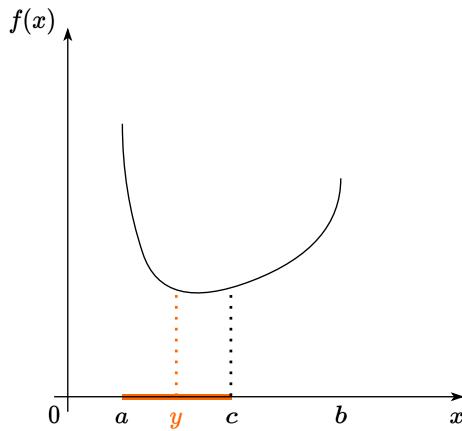


Рисунок 4: Диаграмма метода дихотомии

Длина отрезка на k -ой итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k} (b - a)$$

Для унимодальных функций:

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

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:

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

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.

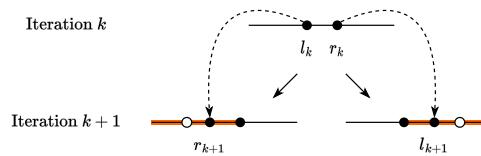


Рисунок 5: Иллюстрация метода золотого сечения

```
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
```

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
```

Сходится сверхлинейно, но метод довольно неустойчивый. Если $f(x)$ не похожа на параболу, гарантий сходимости нет.

26. Условие достаточного убывания для неточного линейного поиска.

💡 Неточный линейный поиск:

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

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:

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

Тогда условием достаточного убывания (Armijo condition) является:

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

Иллюстрация для понимания:

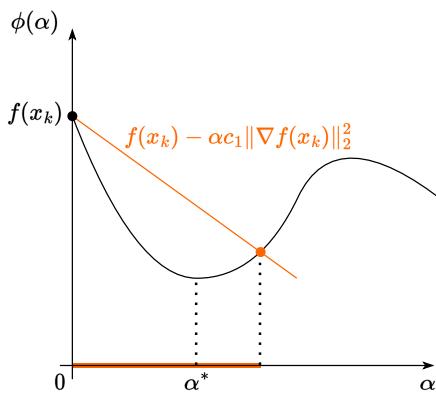


Рисунок 6: Иллюстрация условия достаточного убывания

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:

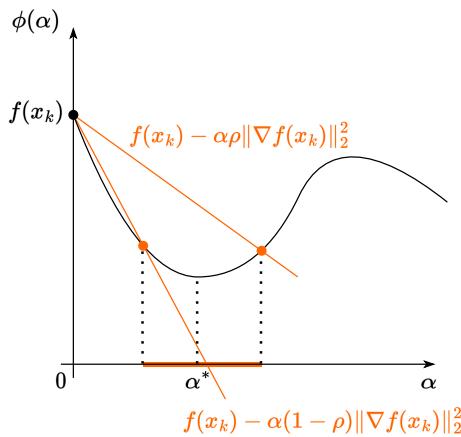


Рисунок 7: Иллюстрация условий Гольдштейна

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:

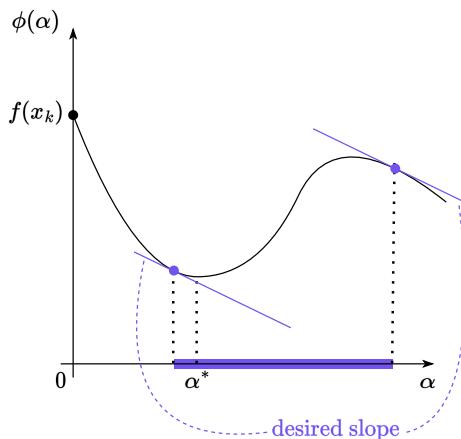


Рисунок 8: Иллюстрация условия ограничения на кривизну

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

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

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

💡

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

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

💡

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

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 :

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

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 :

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

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

💡

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

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

💡

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

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



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

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

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).



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

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).



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

39. Афинное 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.

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

40. 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

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

Выпуклая оболочка – множество всех возможных выпуклых комбинаций элементов множества.

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

41. 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

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

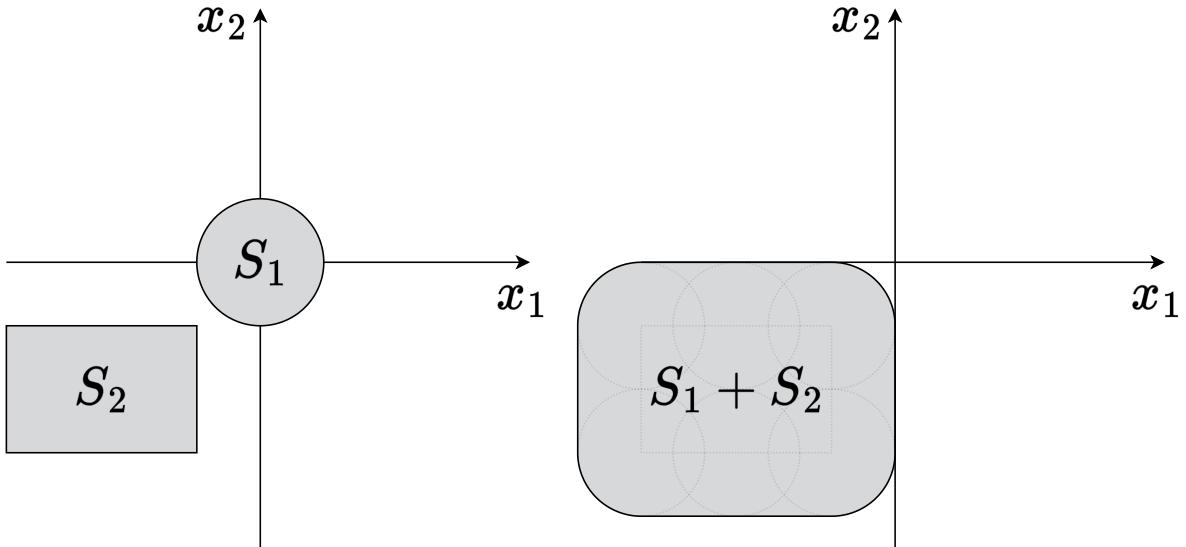
Коническая оболочка - множество всех возможных конических комбинаций элементов множества.

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

42. Сумма Минковского.

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

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



43. 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:

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

44. Выпуклая функция.

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

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

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

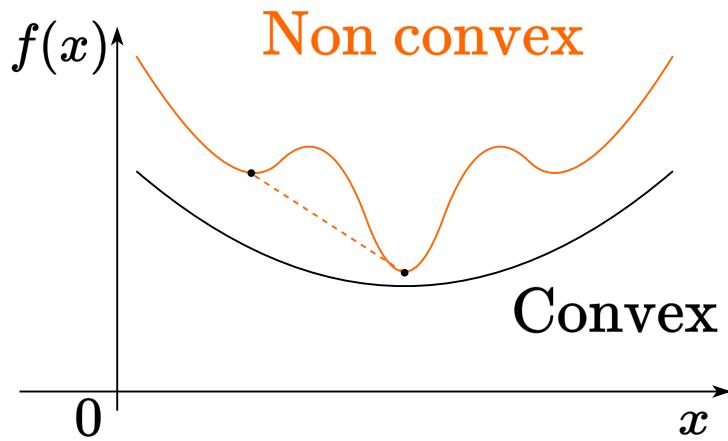


Рисунок 9: Иллюстрация выпуклой функции

45. 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 $f(x)$.

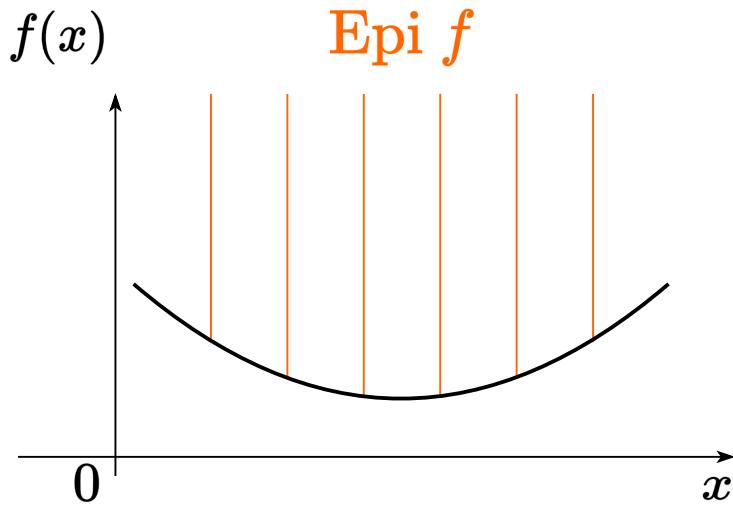


Рисунок 10: Иллюстрация надграфика функции

46. Дифференциальный критерий выпуклости первого порядка.

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

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

47. 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 $x \in \text{int}(S) \neq \emptyset$:

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

48. Связь выпуклости функции и её надграфика.

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

49. μ -сильно выпуклая функция.

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

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

50. 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 $\forall x, y \in S$:

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

51. 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 существует $\mu > 0$

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

52. Любые 2 операции с функциями, сохраняющие выпуклость.

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

53. Необходимые условия локального экстремума.

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

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

54. 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

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

55. Общая задача математического программирования. Функция Лагранжа.



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

Функция Лагранжа:

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

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



Пусть x_* - решение задачи с нулевым зазором двойственности

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

Функция Лагранжа:

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

Тогда найдутся такие векторы λ^* и ν^* , что выполнены условия

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

57. Условие Слейтера.



- 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)
- 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.

58. Задача выпуклого программирования.

💡 Задача выпуклого программирования — это задача оптимизации, в которой целевая функция является выпуклой функцией и область допустимых решений выпукла. В форме ниже функции f_0, \dots, f_m - выпуклые, а функции h_i - афинные.

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

59. Задача линейного программирования. Задача линейного программирования в стандартной форме.

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

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

60. Идея симплекс метода.

💡 Симплекс метод решает следующую задачу:

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

Шаги выполнения симплекс метода:

1. Поиск начальной базисной допустимой точки
2. Проверка оптимальности. Если решение оптимально, то алгоритм завершается, иначе переходим к следующему шагу
3. Замена базиса
4. Повторяем предыдущие два шага до достижения оптимального решения или установления, что задача не имеет допустимого решения

61. 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.

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

💡 Пусть f дифференцируема, зададим искомое направление локального убывания - h - $\|h\| = 1$. Тогда её аппроксимация: $f(x + \alpha h) = f(x) + \alpha \langle \nabla f(x), h \rangle + o(\alpha)$

$$f(x + \alpha h) < f(x) \Rightarrow \alpha \langle \nabla f(x), h \rangle + o(\alpha) < 0.$$

При $\alpha \rightarrow +0$ получаем: $\alpha \langle \nabla f(x), h \rangle \leq 0$

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

$$\langle \nabla f(x), h \rangle \geq -\|\nabla f(x)\| \Rightarrow h = \frac{-\nabla f(x)}{\|\nabla f(x)\|}, \text{ ч.т.д.}$$

63. Метод градиентного спуска.

💡 Решаем задачу минимизации

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

Если f дифференцируема, то тогда для решения этой задачи можно использовать метод градиентного спуска:

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

64. Наискорейший спуск.

💡 Решаем задачу минимизации

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

Если f дифференцируема, то тогда для решения этой задачи можно использовать метод наискорейшего спуска:

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

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

т.е. выбираем наилучший шаг спуска на каждой итерации метода.

65. Как направлены две соседние итерации метода наискорейшего спуска по отношению друг к другу?

💡 Они ортогональны друг другу.

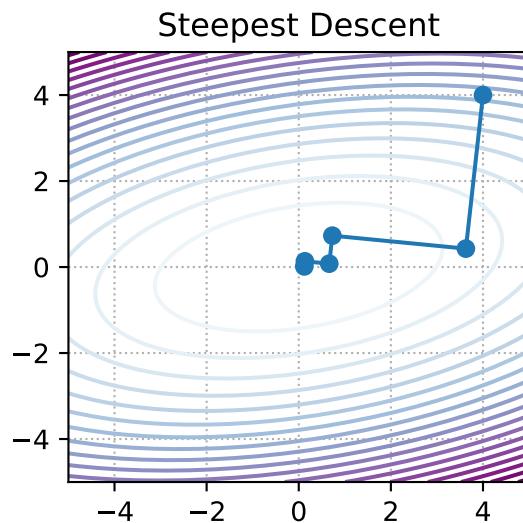


Рисунок 11: Несколько итераций наискорейшего спуска

66. Липшицева парабола для гладкой функции.

💡 Если $f : \mathbb{R}^n \rightarrow \mathbb{R}$ - непрерывно дифференцируема и градиент Липшицев с константой L , то $\forall x, y \in \mathbb{R}^n$:

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

Если зафиксируем $x_0 \in \mathbb{R}^n$, то:

$$\varphi_1(x) = f(x_0) + \langle f(x_0), x - x_0 \rangle - \frac{L}{2} \|x - x_0\|^2$$

$$\varphi_2(x) = f(x_0) + \langle f(x_0), x - x_0 \rangle + \frac{L}{2} \|x - x_0\|^2$$

Это две параболы, и для них верно, что $\varphi_1(x) \leq f(x) \leq \varphi_2(x) \forall x$

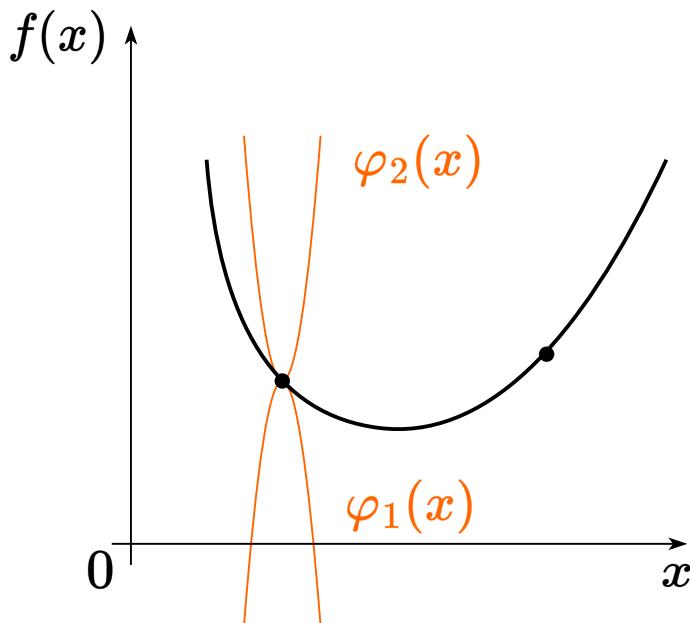


Рисунок 12: Иллюстрация Липшицевых парабол, между которыми зажата гладкая функция. Чаще нас интересует мажорирующая из них.

67. Размер шага наискорейшего спуска для квадратичной функции.

💡 Решаем задачу минимизации методом наискорейшего спуска

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

$$\nabla f = \frac{1}{2}(A + A^T)x - b$$

Из условия $\nabla f(x_{k+1})^T \nabla f(x_k) = 0$ получаем:

$$\alpha_k = \frac{2\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T (A + A^T) \nabla f(x_k)} = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)}.$$

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

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

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

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

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

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

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

💡 $\exists \mu > 0 : \|\nabla f(x)\|^2 \geq 2\mu(f(x) - f^*) \quad \forall x$, где f^* - минимум функции $f(x)$.

72. Сходимость градиентного спуска для сильно выпуклых квадратичных функций. Оптимальные гиперпараметры.

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

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

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

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

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

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

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

73. Связь PL-функций и сильно выпуклых функций.

- 💡 Пусть f μ -сильно выпуклая и дифференцируемая $\Rightarrow f \in \text{PL}$.
Обратное неверно - $f(x) = x^2 + 3\sin^2 x \in \text{PL}$, но не сильно выпуклая (она вообще не выпуклая).

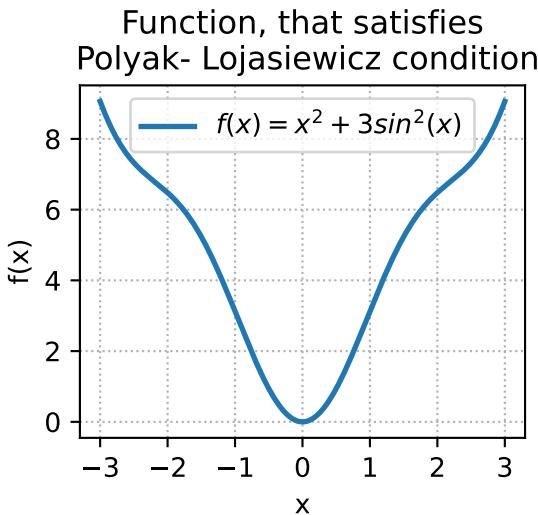


Рисунок 13: Пример невыпуклой PL функции

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

- 💡 Рассмотрим задачу минимизации функции:

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

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

75. 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.

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

💡 $f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{k^2}\right)$

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

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

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



Функция	Неускоренная	Ускоренная
Гладкая и выпуклая	$\mathcal{O} \left(\frac{1}{k^2} \right)$	$\mathcal{O} \left(\frac{1}{k} \right)$
Гладкая и сильно-выпуклая (или PL)	$\mathcal{O} \left((1 - \frac{\mu}{L})^k \right)$	$\mathcal{O} \left((1 - \sqrt{\frac{\mu}{L}})^k \right)$

79. Метод тяжелого шарика (Поляка).

💡 Задача: $f(x) \rightarrow \min_{x \in R^d}$, $f(x)$ - непрерывно дифференцируемая функция

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}), \quad 0 < \beta < 1.$$

80. Понятие локальной и глобальной сходимости численного метода оптимизации.

💡 Локальная сходимость означает, что последовательность итераций сходится к решению x^* внутри некоторого множества. За пределами этого множества сходимости нет. Глобальная сходимость означает, что последовательность итераций сходится к решению x^* вне зависимости от начального приближения.

81. Ускоренный градиентный метод Нестерова для выпуклых гладких функций.

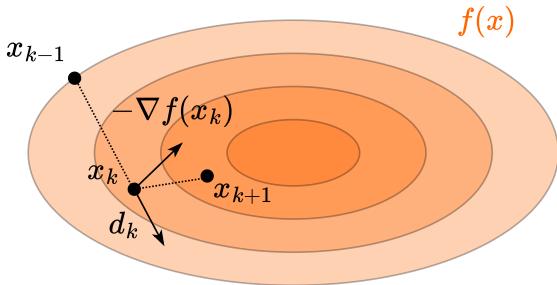
💡 Пусть $f : \mathbb{R}^n \rightarrow \mathbb{R}$ выпуклая и L -гладкая. Ускоренный градиентный метод Нестерова (NAG) имеет вид ($x_0 = y_0, \lambda_0 = 0$):

Градиентный шаг:	$y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$
Экстраполяция:	$x_{k+1} = (1 - \gamma_k)y_{k+1} + \gamma_k y_k$
Вес экстраполяции:	$\lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2}$
Вес экстраполяции:	$\gamma_k = \frac{1 - \lambda_k}{\lambda_{k+1}}$

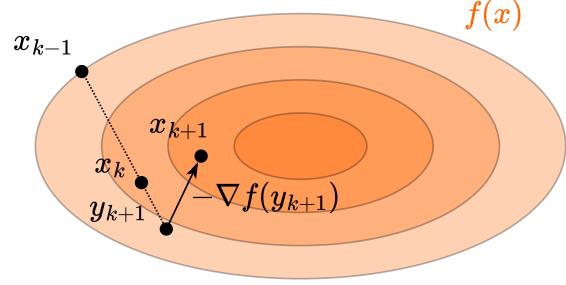
Последовательность $\{f(y_k)\}_{k \in \mathbb{N}}$ сходится к оптимальному значению f^* со скоростью $\mathcal{O}\left(\frac{1}{k^2}\right)$, а именно:

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

Polyak momentum



Nesterov momentum



82. Ускоренный градиентный метод Нестерова для сильно выпуклых гладких функций.

💡 Пусть $f : \mathbb{R}^n \rightarrow \mathbb{R}$ μ -сильно выпуклая и L -гладкая. Ускоренный градиентный метод Нестерова (NAG) имеет вид ($x_0 = y_0, \lambda_0 = 0$):

Градиентный шаг:	$y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$
Экстраполяция:	$x_{k+1} = (1 + \gamma_k)y_{k+1} - \gamma_k y_k$
Вес экстраполяции:	$\gamma_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$

Последовательность $\{f(y_k)\}_{k \in \mathbb{N}}$ сходится к оптимальному значению f^* линейно:

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

83. A -сопряженность двух векторов. A -ортогональность. Скалярное произведение $\langle \cdot, \cdot \rangle_A$.

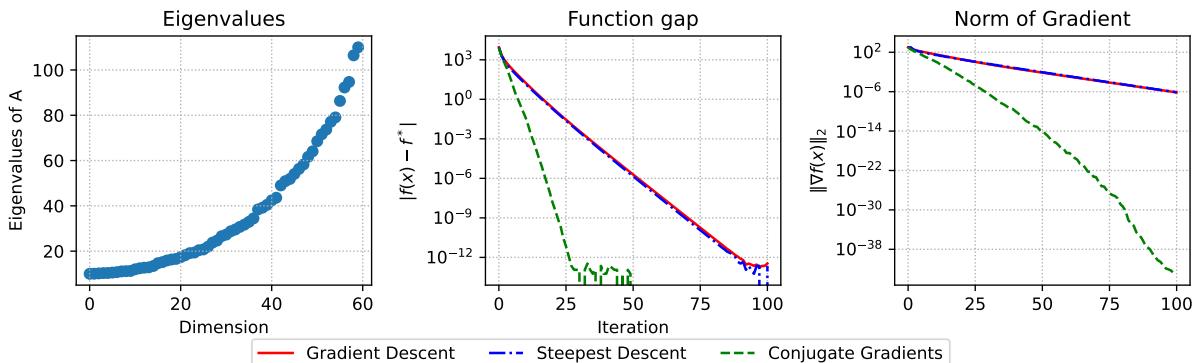
💡 A-ортогональность (сопряженность):

$$x \perp_A y \iff x^T A y = 0.$$

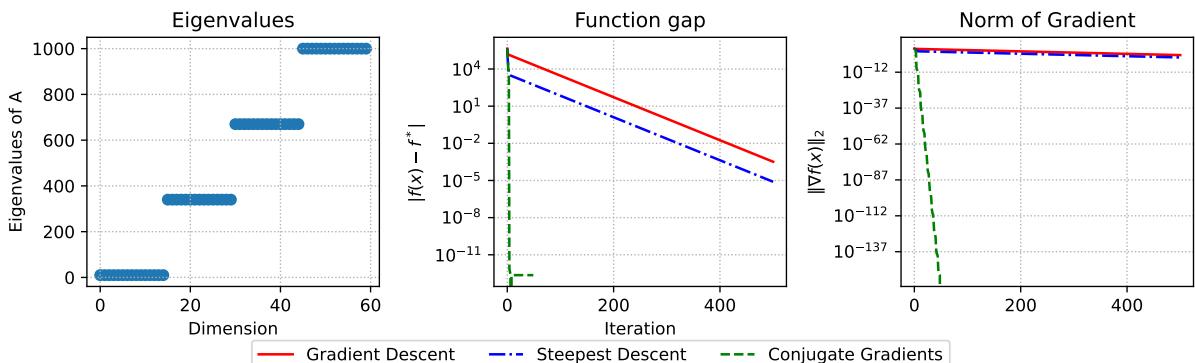
84. Зависимость сходимости метода сопряженных градиентов от спектра матрицы.

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

Strongly convex quadratics. n=60, random matrix.



Strongly convex quadratics. n=60, clustered matrix.



85. Метод Ньютона.

💡 Рассматривается задача минимизации функции с невырожденным гессианом.

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

86. Сходимость метода Ньютона для квадратичной функции.

💡 Метод Ньютона сходится для квадратичной функции за одну итерацию, при условии, что гессиан невырожден. Следует из того, что метода Ньютона минимизирует квадратичную тейлоровскую аппроксимацию функции, которая в случае квадратичной функции совпадает с самой функцией:

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

87. 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 вдобавок, Гессиан M -Липшицев, тогда метод сходится локально к x^* с квадратичной скоростью.

88. 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.

89. Идея квазиньютоновских методов. Метод SR-1.



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

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 .

$$B_{k+1} = B_k + \frac{(\Delta y_k - B_k d_k)(\Delta y_k - B_k d_k)^T}{(\Delta y_k - B_k d_k)^T d_k}, \quad \Delta y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

90. Проекция.



Проекция точки $y \in \mathbb{R}^n$ на множество $S \subseteq \mathbb{R}^n$ это точка $\text{proj}_S(y) \in S$:

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

91. Достаточное условие существования проекции точки на множество.



Если $S \subseteq \mathbb{R}^n$ – замкнутое множество, тогда проекция на множество S существует для любой точки.

92. Достаточное условие единственности проекции точки на множество.



Если $S \subseteq \mathbb{R}^n$ – замкнутое выпуклое множество, тогда проекция на множество S единствена для каждой точки.

93. Метод проекции градиента.

💡 Рассматривается задача $f(x) \rightarrow \min_{x \in S}$, где $S \subseteq \mathbb{R}^n$. Метод проекции градиента — это метод оптимизации с проекцией на бюджетное множество S :

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

где α_k — learning rate.

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

💡 Проекция $\text{proj}_S(x)$ точки x на выпуклое множество S удовлетворяет:

$$\langle x - \text{proj}_S(x), y - \text{proj}_S(x) \rangle \leq 0 \quad \forall y \in S.$$

95. Проекция как нерастягивающий оператор.

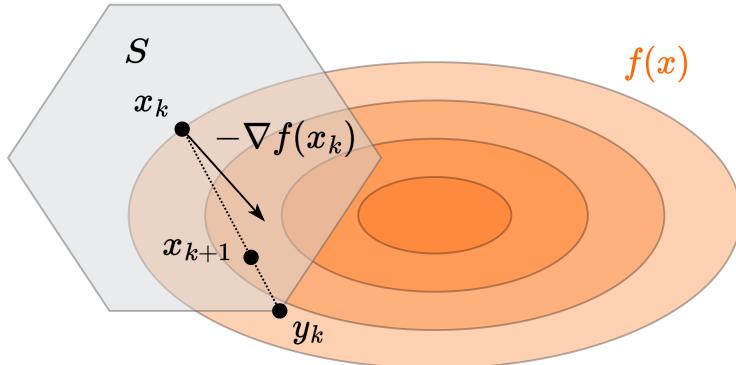
💡 Проекция на выпуклое множество S является нерастягивающим оператором:

$$\|\text{proj}_S(x) - \text{proj}_S(y)\| \leq \|x - y\| \quad \forall x, y.$$

96. Метод Франк-Вульфа.

💡 Рассматриваем задачу $f(x) \rightarrow \min_{x \in S}$. Метод Франк-Вульфа имеет вид:

$$\begin{aligned} y_k &= \arg \min_{x \in S} f_{x_k}^I(x) = \arg \min_{x \in S} \langle \nabla f(x_k), x \rangle \\ x_{k+1} &= \gamma_k x_k + (1 - \gamma_k) y_k \end{aligned}$$



где γ_k — гиперпараметр.

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

💡 Для гладких выпуклых функций метод проекции градиента имеет сходимость порядка $\mathcal{O}\left(\frac{1}{k}\right)$, где k — число итераций. То есть сходимость такая же, как и для безусловной задачи, но стоимость итерации может быть выше из-за проекции.

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

💡 Для гладких сильно выпуклых функций метод проекции градиента имеет линейную сходимость порядка $\mathcal{O}\left((1 - \frac{\mu}{L})^k\right)$, где k — число итераций. То есть сходимость такая же, как и для безусловной задачи, но стоимость итерации может быть выше из-за проекции.

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

💡 Метод Франк-Вульфа для гладких выпуклых функций имеет сходимость порядка $\mathcal{O}\left(\frac{1}{k}\right)$, где k — число итераций.

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

💡 Для гладких сильно выпуклых функций метод Франк-Вульфа имеет сходимость порядка $\mathcal{O}\left(\frac{1}{k}\right)$, где k — число итераций.

101. 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.

102. Субградиент. Субдифференциал.

💡 Субградиент функции f в точке x — это вектор g , удовлетворяющий условию:

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

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

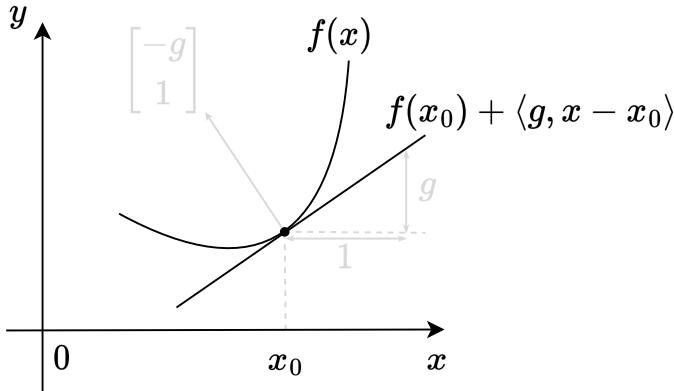


Рисунок 14: Субдифференциал функции ReLU.

103. Как считать субдифференциал поточечного максимума выпуклых функций.

💡 Пусть $f_i(x)$ — выпуклые функции на открытом выпуклом множестве $S \subseteq \mathbb{R}^n$, $x_0 \in S$, и поточечный максимум определяется как $f(x) = \max_i f_i(x)$. Тогда:

$$\partial_S f(x_0) = \mathbf{conv} \left\{ \bigcup_{i \in I(x_0)} \partial_S f_i(x_0) \right\}, \quad I(x) = \{i \in [1 : m] : f_i(x) = f(x)\}$$

104. Субградиентный метод.

💡 Субградиентный метод используется для минимизации выпуклых функций, которые могут быть негладкими. Итерационная формула метода:

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

где $g_k \in \partial f(x_k)$ — субградиент функции f в точке x_k , α_k — шаг метода на k -й итерации.

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

💡 $f_k^{\text{best}} - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$

106. Какому условию должна удовлетворять стратегия выбора шага, чтобы субградиентный метод сходился для выпуклых Липшицевых функций?

?

$$\sum_{i=1}^k \alpha_i = \infty, \quad \sum_{i=1}^k \alpha_i^2 < \infty$$

107. Характер сходимости субградиентного метода для негладких сильно выпуклых Липшицевых функций в терминах \mathcal{O} от числа итераций метода. Стратегия выбора шага.

?

$$f_k^{\text{best}} - f^* \sim \mathcal{O}\left(\frac{1}{k}\right), \alpha_k = \frac{2}{\mu(k+1)}$$

108. Метод стохастического градиентного спуска.

?

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

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

где i_k - случайно выбранный индекс. Если $\mathbb{P}(i_k = i) = \frac{1}{n}$, то $\mathbb{E}[\nabla f_{i_k}(x)] = \nabla f(x)$

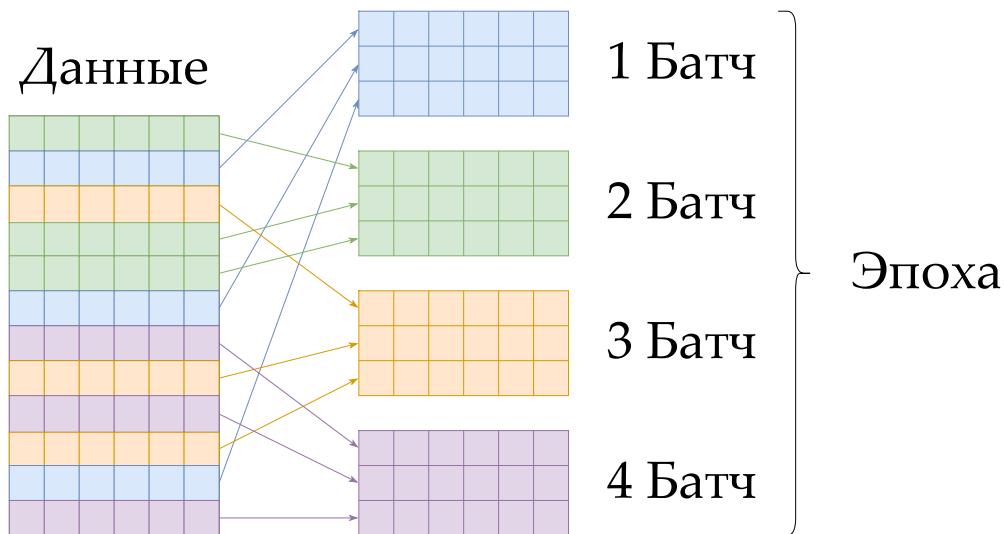
109. Идея мини-батча для метода стохастического градиентного спуска. Эпоха.

?

Разделим данные размера N на k мини-батчей (выборок) размера B_k , на каждой итерации посчитаем градиент мини-батча с использованием параллелизма. За $\frac{N}{k}$ итераций пройдёмся по всей выборке. Эпоха - набор k итераций с батчем размера $B_k = \frac{N}{k}$.

$$x_{k+1} = x_k - \frac{1}{|B_k|} \sum_{i \in B_k} -\text{шаг мини-батча}.$$

С увеличением размера мини-батча время на эпоху уменьшается до тех пор, пока нам хватает памяти (в случае наличия параллелизма).



110. Характер сходимости стохастического градиентного спуска для гладких выпуклых функций в

терминах \mathcal{O} от числа итераций метода.

💡 f - гладкая и выпуклая $\Rightarrow \mathcal{O}\left(\frac{1}{\varepsilon^2}\right), \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$

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

💡 $f \in \text{PL} \Rightarrow \mathcal{O}\left(\frac{1}{k}\right), \mathcal{O}\left(\frac{1}{\varepsilon}\right)$

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

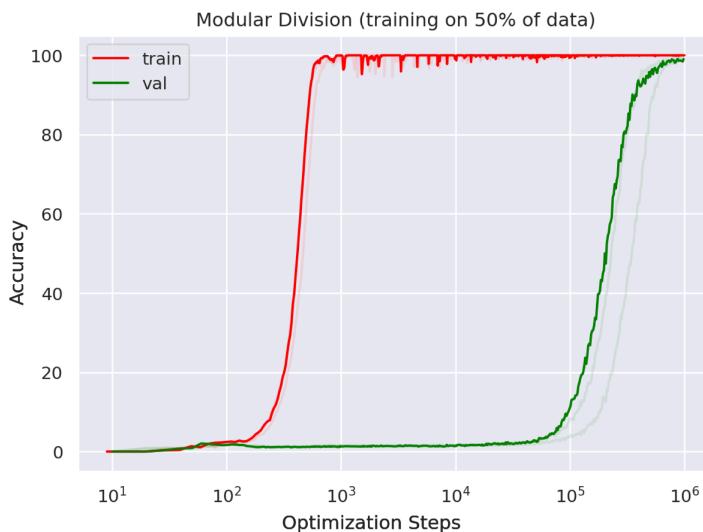
💡 Пусть $\min_{x \in \mathbb{R}^p} f(x) = \min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n f_i(x)$ при использовании стохастического градиентного спуска с постоянным шагом α

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

имеем следующую оценку $\mathbb{E}[f(x_{k+1}) - f^*] \leq (1 - 2\alpha\mu)^k [f(x_0) - f^*] + \frac{L\sigma^2\alpha}{4\mu}$. Характер сходимости - линейный до некоторого шара несходимости, в котором будут происходить осцилляции и сходимости не будет.

113. 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.



114. 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.

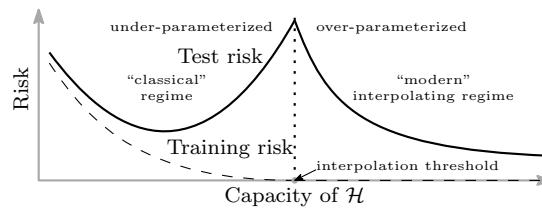


Рисунок 15: Иллюстрация зависимости обобщающей способности модели от размера.

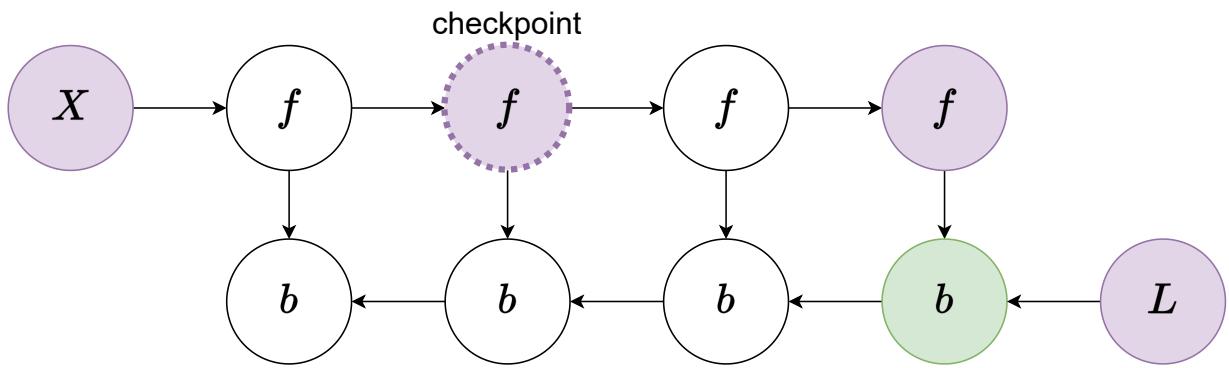
115. Взрыв/Затухание градиентов при обучении глубоких нейронных сетей.

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: $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z^{(n)}} \cdot \frac{\partial z^{(n)}}{\partial z^{(n-1)}} \dots \frac{\partial z^{(2)}}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial W}$, где $z^{(i)}$ - активации 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.

116. 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.



2. Таким образом, в случае $\alpha = 1$ последовательность $(r_k)_{k=m}^{\infty}$ не может иметь линейной сходимости в соответствии с приведенным выше результатом (доказано от противного). Тем не менее, $(r_k)_{k=m}^{\infty}$ сходится к нулю, поэтому она должна сходиться сублинейно.
3. Теперь рассмотрим случай $0 \leq \alpha < 1$. Пусть $\varepsilon > 0$ — произвольное число такое, что $\alpha + \varepsilon < 1$. Согласно свойствам \limsup , существует $N \geq m$ такое, что $r_k^{1/k} \leq \alpha + \varepsilon$ для всех $k \geq N$. Следовательно, $r_k \leq (\alpha + \varepsilon)^k$ для всех $k \geq N$. Таким образом, $(r_k)_{k=m}^{\infty}$ сходится линейно с параметром $\alpha + \varepsilon$ (не важно, что неравенство выполняется только начиная с числа N). Учитывая произвольность ε , это означает, что константа линейной сходимости $(r_k)_{k=m}^{\infty}$ не превышает α . Поскольку, как показано выше, константа линейной сходимости не может быть меньше α , это означает, что константа линейной сходимости $(r_k)_{k=m}^{\infty}$ точно равна α .
4. Наконец, покажем, что случай $\alpha > 1$ невозможен. Действительно, предположим, что $\alpha > 1$. Тогда из определения \limsup следует, что для любого $N \geq m$ существует $k \geq N$ такое, что $r_k^{1/k} \geq 1$, и, в частности, $r_k \geq 1$. Но это означает, что r_k имеет подпоследовательность, ограниченную от нуля. Следовательно, $(r_k)_{k=m}^{\infty}$ не может сходиться к нулю, что противоречит условию.

2. Метод дихотомии и золотого сечения для унимодальных функций. Скорость сходимости.

i Методы локализации решения для скалярной минимизации. Сходятся линейно.

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:

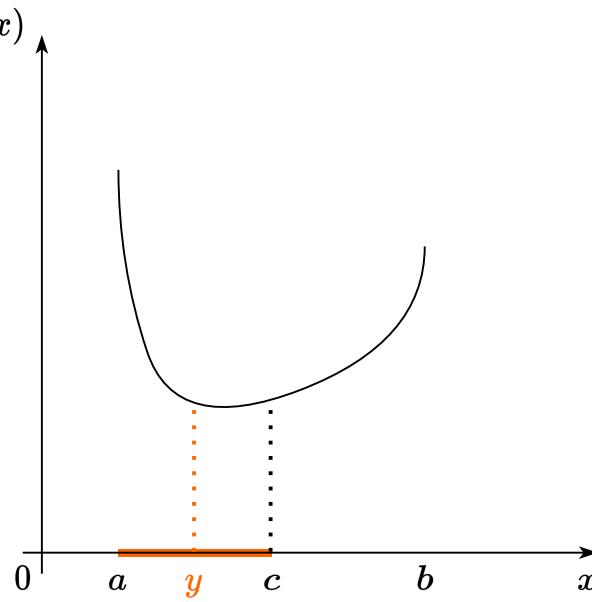


Рисунок 16: Алгоритм дихотомии

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

Сходимость метода дихотомии

Длина отрезка на k -ой итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k} (b - a)$$

Если будем выбирать середину отрезка как выход k -ой итерации:

$$|x_k - x_*| \leq \frac{\Delta_k}{2}$$

Подставим полученное ранее выражение для длины отрезка:

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

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

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 $N = 2k$:

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

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

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:

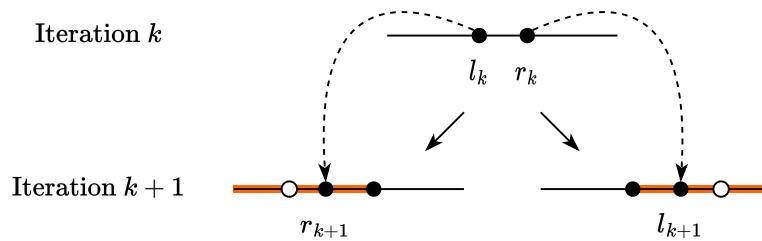


Рисунок 17: Золотое сечение

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):

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

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.

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

i Для функции $L(w_1, w_2, \dots, w_d)$ вычислительный граф представляет собой ориентированный ациклический граф с двумя типами вершин: w_1, w_2, \dots, w_d и f_1, f_2, \dots . Здесь w_1, w_2, \dots, w_d обозначают входные переменные, а v_1, v_2, \dots обозначают 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 характерно для функции, используемой в обучении нейросети, где может быть множество вершин-стоков L_1, L_2, \dots .

$$L(w_1, w_2) = w_2 \log w_1 + \sqrt{w_2 \log w_1}$$

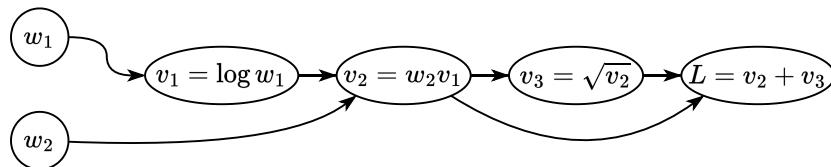


Рисунок 18: Вычислительный график

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

Forward mode

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

- Вычисляется v_i как функция 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 правила производной сложной функции:

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

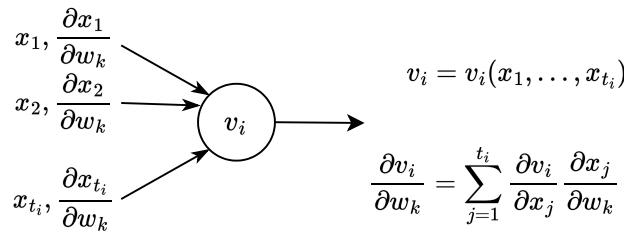


Рисунок 19: Forward mode

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 $O(dT)$.

Backward mode

Forward pass

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

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

Backward pass

Обрабатывая вершины в обратном порядке топологического упорядочения:

- Вычисляется производная \dot{v}_i , используя правило дифференцирования сложной функции в обратном порядке и информацию от всех его потомков (выходов) (x_1, \dots, x_{t_i}) , используя предподсчитанные значения на forward pass:

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

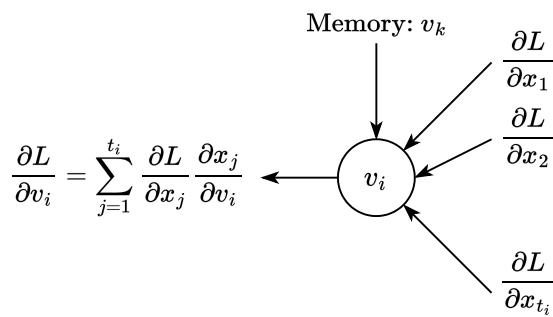
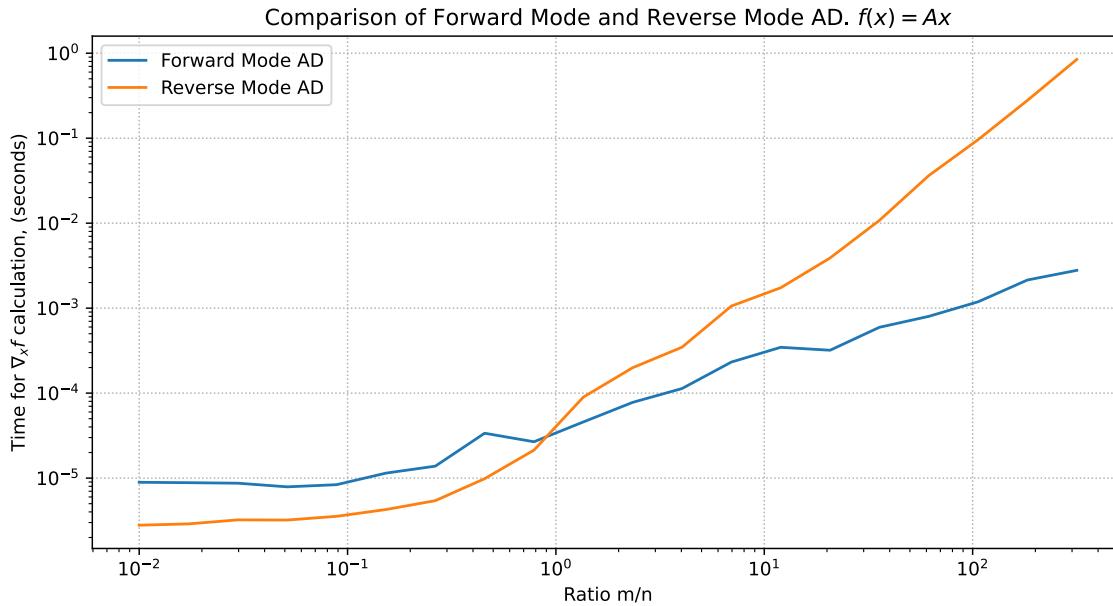


Рисунок 20: Backward mode

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



4. Выпуклость надграфика как критерий выпуклости функции.

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

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 $\mu \in \mathbb{R}$, $x \in S$).

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.

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 .

5. Дифференциальный критерий сильной выпуклости первого порядка.

i 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.

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.

6. Дифференциальный критерий сильной выпуклости второго порядка.

i 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:

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

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.

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.

7. Необходимые условия безусловного экстремума.

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

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

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.

8. Достаточные условия безусловного экстремума.

i 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 .

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.

9. Симплекс метод для задачи линейного программирования с ограничениями в форме неравенств. Двухфазный симплекс метод.

i Решаемая задача:

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

Идея 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 сходимости

Идея двухфазного симплекс метода:

1. Сформулировать вспомогательную задачу линейного программирования, решение которой даст начальную базисную допустимую точку для исходной задачи.
2. Угадать начальную базисную допустимую точку для вспомогательной задачи и решить её (симплекс методом).
3. Использовать решение вспомогательной задачи как начальную базисную допустимую точку для исходной задачи.

Задача линейного программирования с ограничениями в форме неравенств:

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

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

Шаги выполнения симплекс метода:

1. Поиск начальной базисной допустимой точки:

- Выберем начальную базисную точку (она является решением системы $A_{\mathcal{B}}x_0 = b_{\mathcal{B}}$, где \mathcal{B} – базис размера n , а матрица A обычно имеет больше n ограничений) допустимую ($Ax_0 \leq b$) точку x_0 (искать ее будем через двухфазный симплекс-метод). $A_{\mathcal{B}} \in \mathbb{R}^{n \times n}$ – подматрица матрицы A с базисными строками, $b_{\mathcal{B}} \in \mathbb{R}^n$ – подвектор вектора b с базисными строками.

2. Проверка оптимальности:

- Разложение вектора c в данном базисе \mathcal{B} с коэффициентами $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^\top A_{\mathcal{B}} = c^\top \quad \text{или} \quad \lambda_{\mathcal{B}}^\top = c^\top A_{\mathcal{B}}^{-1}$$

- Если все компоненты $\lambda_{\mathcal{B}}$ неположительны, текущий базис является оптимальным. Иначе далее меняем вершину симплекса.

3. Определение переменной для удаления из базиса:

- Если в разложении $\lambda_{\mathcal{B}}$ есть положительные координаты, продолжаем оптимизацию. Пусть $\lambda_{\mathcal{B}}^k > 0$. Необходимо исключить k из базиса. Рассчитаем направляющий вектор d , идя вдоль которого изменим вершину следующим образом: во-первых, для всех векторов ограничений из базиса, которые мы оставляем, подбираемое направление должно быть им ортогонально (т.е. движение вдоль него не будет нарушать остальные ограничения), и, во-вторых, вдоль него значение, связанное с нашим ограничением, должно убывать:

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

4. Вычисление шага вдоль выбранного направления d :

- Для всех $j \notin \mathcal{B}$ считаем шаг:

$$\mu_j = \frac{b_j - a_j^\top x_{\mathcal{B}}}{a_j^\top d}$$

- Новая вершина, которую добавим в базис:

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

5. Обновление базиса:

- Обновляем базис и текущее решение:

$$\begin{aligned} \mathcal{B}' &= \mathcal{B} \setminus \{k\} \cup \{t\}, \\ x_{\mathcal{B}'} &= x_{\mathcal{B}} + \mu_t d = A_{\mathcal{B}'}^{-1} b_{\mathcal{B}'} \end{aligned}$$

- Изменение базиса приводит к уменьшению значения целевой функции:

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

6. Повторение:

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

Нахождение начальной базисной допустимой точки

Целью является решение следующей задачи:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & c^\top x \\ \text{s.t. } & Ax \leq b \end{aligned} \tag{1}$$

Предложенный алгоритм требует начальной базисной допустимой точки и соответствующего базиса.

Начнем с переформулирования задачи:

$$\begin{aligned} & \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\ \text{s.t. } & Ay - Az \leq b \\ & y \geq 0, z \geq 0 \end{aligned} \tag{2}$$

Имея решение задачи 2, решение задачи 1 может быть однозначно получено и наоборот:

$$x = y - z \Leftrightarrow y_i = \max(x_i, 0), \quad z_i = \max(-x_i, 0)$$

Теперь мы попытаемся сформулировать новую задачу ЛП, решение которой будет базисной допустимой точкой для задачи 2. Это означает, что мы сначала запустим симплекс-метод для задачи 2, а затем запустим задачу 1 с известной начальной точкой. Обратите внимание, что базисная допустимая точка для задачи 2 должна быть легко вычислена.

$$\begin{aligned} & \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\ \text{s.t. } & Ay - Az \leq b \\ & y \geq 0, z \geq 0 \end{aligned} \tag{Phase-2 (Main LP)}$$

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

- Если задача 2 имеет допустимое решение, то оптимум задачи 2 равен нулю (т.е. все переменные ξ_i равны нулю).

Доказательство: тривиальная проверка.

- Если оптимум задачи 2 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для задачи 2.

Доказательство: тривиальная проверка.

- Теперь мы знаем, что если мы можем решить задачу 2, то мы либо найдем начальную точку для симплекс-метода в исходной задаче (если все переменные ξ_i равны нулю), либо проверим, что исходная задача не имеет допустимого решения (если все переменные ξ_i не равны нулю).
- Но как решить задачу 2? Она имеет базисную допустимую точку (задача имеет $2n + m$ переменных и точка ниже гарантирует, что $2n + m$ неравенств выполняются как равенства (активны)).

$$z = 0 \quad y = 0 \quad \xi_i = \max(0, -b_i)$$

10. Теорема сходимости градиентного спуска для гладких PL функций.

i Рассмотрим задачу

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

и предположим, что f удовлетворяет условию Поляка-Лоясиевича с константой μ и L -гладкости, для некоторых $L \geq \mu > 0$. Пусть $(x_k)_{k \in \mathbb{N}}$ - последовательность, созданная градиентным спуском с постоянным шагом α , $0 < \alpha \leq \frac{1}{L}$. Тогда имеется линейная сходимость:

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

1. Используя L -гладкость, вместе с правилом обновления алгоритма, можно записать:

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

В последнем неравенстве использовали предположение о шаге $0 < \alpha L \leq 1$.

2. Используя свойство Поляка-Лоясиевича, можно записать:

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

3. Вычитая f^* с обеих сторон и используя рекурсию, получаем:

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

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

i

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

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

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

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

$$\nabla f(x) = Ax - b \stackrel{\nabla f(x^*)=0}{\Rightarrow} Ax^* = b$$

Тогда шаг градиентного спуска имеет вид

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

Найдем α^* . Воспользуемся $A = Q\Lambda Q^T$, где $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}$, $Q = [q_1, \dots, q_n]$, λ_i, q_i - собственное значение и собственный вектор соответственно.

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

$$\begin{aligned}x_{k+1} - x^* &= (I - \alpha A)(x_k - x^*) \\x_{k+1} - x^* &= (I - \alpha Q \Lambda Q^T)(x_k - x^*) | \cdot Q^T \\Q^T(x_{k+1} - x^*) &= (Q^T - \alpha \Lambda Q^T)(x_k - x^*) = (I - \alpha \Lambda)Q^T(x_k - x^*)\end{aligned}$$

Замена: $\tilde{x} = Q^T(x - x^*) \Rightarrow \tilde{x}_{k+1} = (I - \alpha \Lambda)\tilde{x}_k \Leftrightarrow \tilde{x}_i^{(k+1)} = (1 - \alpha \lambda_i)\tilde{x}_i^{(k)} i = \overline{1, d}$
 $\lambda_{\min} = \mu, \quad \lambda_{\max} = L$

Сходимость есть $\Leftrightarrow \max_i |1 - \alpha \lambda_i| < 1$

$$\left\{ \begin{array}{l} |1 - \lambda \mu| < 1 \Rightarrow 1 - \lambda \mu < 1 \Rightarrow \alpha > 0 \\ \alpha \mu - 1 < 1 \Rightarrow \alpha < \frac{2}{\mu} \\ |1 - \alpha L| < 1 \Rightarrow 1 - \alpha L < 1 \Rightarrow \alpha > 0 \\ \alpha L - 1 < 1 \Rightarrow \alpha < \frac{2}{L} \end{array} \right\} \Rightarrow \alpha < \frac{2}{L}$$

Радиус сходимости $\rho = \max(|1 - \alpha \mu|, |1 - \alpha L|)$ и $\rho \rightarrow \min \Leftrightarrow \alpha^* L - 1 = 1 - \alpha^* \mu \Rightarrow \alpha^* = \frac{2}{\mu + L}$ и
 $\rho^* = \frac{L - \mu}{L + \mu}$

Итого получаем, что для градиентного спуска выполняется $f(x_k) - f^* \leq \left(1 - \frac{\mu}{\mu + L}\right)^k (f(x_0) - f^*)$.