Вспоминаем линейную алгебру. Скорости сходимости.

Даня Меркулов

1 Вспоминаем линейную алгебру

1.1 Векторы и матрицы

Мы будем считать, что все векторы являются столбцами по умолчанию. Пространство векторов длины n обозначается $\mathbb{R}^n$, а пространство матриц размера $m \times n$ с вещественными элементами обозначается $\mathbb{R}^{m\times n}$. То есть [1]:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \quad x \in \mathbb{R}^n, x_i \in \mathbb{R} \tag{1}$$

Аналогично, если $A \in \mathbb{R}^{m\times n}$ мы обозначаем транспонирование как $A^T \in \mathbb{R}^{n\times m}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \quad A \in \mathbb{R}^{m\times n}, a_{ij} \in \mathbb{R}$$

Мы будем писать $x \geq 0$ и $x \neq 0$ для обозначения покомпонентных неравенств

Матрица A называется симметричной, если $A = A^T$. Обозначается как $A \in \mathbb{S}^n$ (множество квадратных симметричных матриц размерности n). Заметим, что только квадратная матрица может быть симметричной по определению.

Матрица $A \in \mathbb{S}^n$ называется **положительно (отрицательно) определенной**, если для всех $x \neq 0$: $x^T Ax > (<)0$. Обозначается как $A \succ (\prec)0$. Множество таких матриц обозначается как $\mathbb{S}^n_{++}(\mathbb{S}^n_{--})$

Матрица $A \in \mathbb{S}^n$ называется **положительно (отрицательно) полуопределенной**, если для всех x : $x^T Ax \geq (\leq)0$. Обозначается как $A \succeq (\preceq)0$. Множество таких матриц обозначается как $\mathbb{S}^n_{+}(\mathbb{S}^n_{-})$

> **Question**
>
> Верно ли, что положительно определенная матрица имеет все положительные элементы?

[1] Подробный вводный курс по прикладной линейной алгебре можно найти в книге Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares - книга от Stephen Boyd & Lieven Vandenberghe, которая указана в источнике. Также полезен материал по линейной алгебре приведенный в приложении A книги Numerical Optimization by Jorge Nocedal Stephen J. Wright.

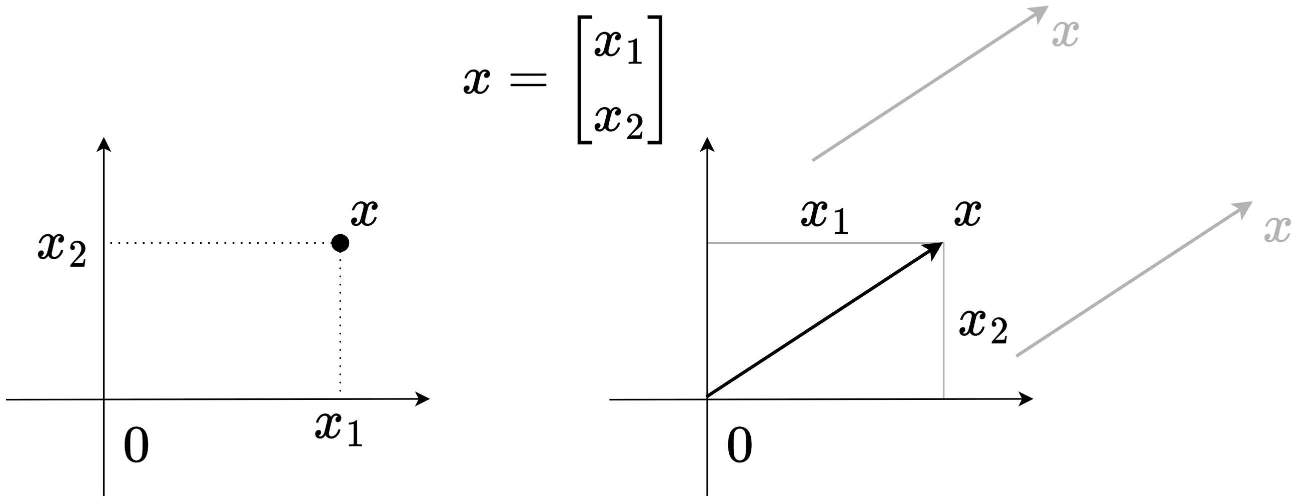


Рисунок 1: Эквивалентные представления вектора

> **Question**
>
> Верно ли, что если матрица симметрична, то она должна быть положительно определенной?

> **Question**
>
> Верно ли, что если матрица положительно определена, то она должна быть симметричной?

1.2 Матричное умножение (matmul)

Пусть A - матрица размера $m \times n$, а B - матрица размера $n \times p$, тогда их произведение AB равно:

$$C = AB$$

Тогда C - матрица размера $m \times p$, элемент (i, j) которой равен:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

Эта операция в наивной форме требует $\mathcal{O}(n^3)$ арифметических операций, где n обычно считается наибольшей размерностью матриц.

> **Question**
>
> Возможно ли умножить две матрицы быстрее, чем за $\mathcal{O}(n^3)$? Как насчет $\mathcal{O}(n^2)$, $\mathcal{O}(n)$?

1.3 Умножение матрицы на вектор (matvec)

Пусть A - матрица размера $m \times n$, а x - вектор длины n, тогда i-й элемент произведения Ax равен:

$$z = Ax$$

равен:

$$z_i = \sum_{k=1}^{n} a_{ik} x_k$$

Эта операция в наивной форме требует $\mathcal{O}(n^2)$ арифметических операций, где n обычно считается наибольшей размерностью входов.

Отметим, что:

- $C = AB \quad C^T = B^T A^T$
- $AB \neq BA$
- $e^A = \sum\limits_{k=0}^{\infty} \frac{1}{k!} A^k$
- $e^{A+B} \neq e^A e^B$ (но если A и B коммутируют, то есть $AB = BA$, то $e^{A+B} = e^A e^B$)
- $\langle x, Ay \rangle = \langle A^T x, y \rangle$

1.4 Нормы

Норма - это **количественная мера малости вектора** и обычно обозначается как $\|x\|$.

Норма должна удовлетворять определенным свойствам:

1. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$
2. $\|x + y\| \leq \|x\| + \|y\|$ (неравенство треугольника)
3. Если $\|x\| = 0$, то $x = 0$

Расстояние между двумя векторами определяется как

$$d(x, y) = \|x - y\|.$$

Наиболее широко используемой нормой является **Евклидова норма**:

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

которая соответствует расстоянию в нашей реальной жизни. Если векторы имеют комплексные элементы, мы используем их модуль. Евклидова норма, или 2-норма, является подклассом важного класса p-норм:

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

1.5 p-норма вектора

Существуют два очень важных частных случая. Бесконечность-норма, или норма Чебышева, определяется как максимальное абсолютное значение элемента вектора:

$$\|x\|_\infty = \max_i |x_i|$$

l_1 норма (или **манхэттенское расстояние**) определяется как сумма модулей элементов вектора x:

$$\|x\|_1 = \sum_i |x_i|$$

l_1 норма играет очень важную роль: она все связана с методами **compressed sensing**, которые появились в середине 00-х как одна из популярных тем исследований. Код для изображения ниже доступен *здесь:*. Также посмотрите *это* видео.

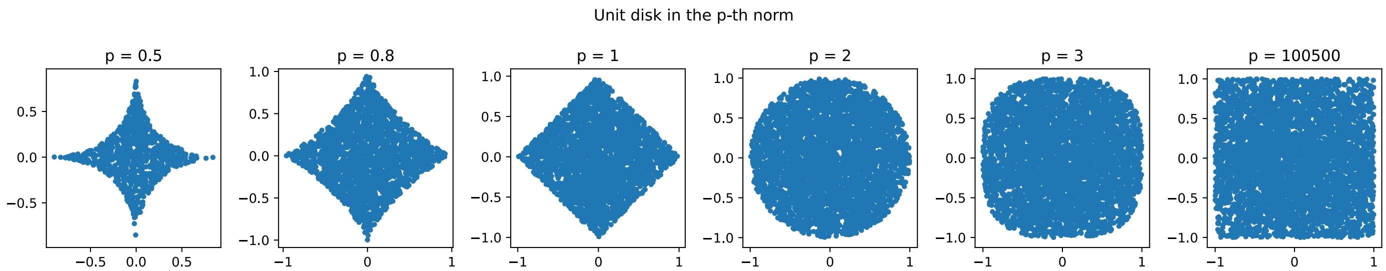


Рисунок 2: Шары в разных нормах на плоскости

1.6 Матричные нормы

В некотором смысле между матрицами и векторами нет большой разницы (вы можете векторизовать матрицу), и здесь появляется самая простая матричная норма **Фробениуса**:

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

Спектральная норма, $\|A\|_2$ является одной из наиболее широко используемых матричных норм (наряду с нормой Фробениуса).

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2},$$

Она не может быть вычислена непосредственно из элементов с помощью простой формулы, как в случае нормы Фробениуса, однако, существуют эффективные алгоритмы для ее вычисления. Она напрямую связана с **сингулярным разложением** (SVD) матрицы. Для неё справедливо:

$$\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_{\max}(A^T A)}$$

где $\sigma_1(A)$ - наибольшее сингулярное значение матрицы A.

1.7 Скалярное произведение

Стандартное **скалярное произведение** между векторами x и y из $\mathbb{R}^n$ равно:

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

Здесь x_i и y_i - i-ые компоненты соответствующих векторов.

> **Example**
>
> Докажите, что вы можете переставить матрицу внутри скалярного произведения с транспонированием: $\langle x, Ay \rangle = \langle A^T x, y \rangle$ и $\langle x, yB \rangle = \langle xB^T, y \rangle$

1.8 Скалярное произведение матриц

Стандартное **скалярное произведение** между матрицами X и Y из $\mathbb{R}^{m\times n}$ равно:

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

> **Question**
>
> Существует ли связь между нормой Фробениуса $\| \cdot \|_F$ и скалярным произведением между матрицами $\langle \cdot, \cdot \rangle$?

1.9 Собственные вектора и собственные значения

Число λ является собственным значением квадратной матрицы A размера $n \times n$, если существует ненулевой вектор q такой, что

$$Aq = \lambda q.$$

Вектор q называется собственным вектором матрицы A. Матрица A невырожденная, если ни одно из её собственных значений не равно нулю. Собственные значения симметричных матриц являются вещественными числами, в то время как несимметричные матрицы могут иметь комплексные собственные значения. Если матрица положительно определена и симметрична, то все её собственные значения являются положительными вещественными числами.

1.10 Собственные вектора и собственные значения

Theorem

$$A \succeq (\succ) 0 \Leftrightarrow \text{все собственные значения } A \geq (>) 0$$

Proof

1. $\rightarrow$ Предположим, что некоторое собственное значение λ отрицательно, и пусть x обозначает соответствующий собственный вектор. Тогда

$$Ax = \lambda x \rightarrow x^T Ax = \lambda x^T x < 0$$

что противоречит условию $A \succeq 0$.

2. $\leftarrow$ Для любой симметричной матрицы мы можем выбрать набор собственных векторов $v_1, \dots, v_n$, которые образуют ортонормированный базис в $\mathbb{R}^n$. Возьмем любой вектор $x \in \mathbb{R}^n$.

$$\begin{aligned} x^T Ax &= (\alpha_1 v_1 + \ldots + \alpha_n v_n)^T A(\alpha_1 v_1 + \ldots + \alpha_n v_n) \\ &= \sum \alpha_i^2 v_i^T Av_i = \sum \alpha_i^2 \lambda_i v_i^T v_i \geq 0 \end{aligned}$$

Здесь мы использовали тот факт, что $v_i^T v_j = 0$, для $i \neq j$.

1.11 Спектральное разложение (eigendecomposition)

Пусть $A \in S_n$, т.е. A - вещественная симметричная матрица размера $n \times n$. Тогда A может быть разложена как

$$A = Q\Lambda Q^T,$$

где $Q \in \mathbb{R}^{n \times n}$ ортогональная, т.е. удовлетворяет $Q^T Q = I$, и $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Вещественные числа λ_i являются собственными значениями A и являются корнями характеристического полинома $\det(A - \lambda I)$. Столбцы Q образуют ортонормированный набор собственных векторов A. Такое разложение называется спектральным. [2]

Мы обычно упорядочиваем вещественные собственные значения как $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Мы используем обозначение $\lambda_i(A)$ для обозначения i-го наибольшего собственного значения $A \in S$. Мы обычно пишем наибольшее или максимальное собственное значение как $\lambda_1(A) = \lambda_{\max}(A)$, и наименьшее или минимальное собственное значение как $\lambda_n(A) = \lambda_{\min}(A)$.

1.12 Собственные значения

Наибольшее и наименьшее вещественныесобственные значения удовлетворяют

$$\lambda_{\min}(A) = \inf_{x \neq 0} \frac{x^T Ax}{x^T x}, \qquad \lambda_{\max}(A) = \sup_{x \neq 0} \frac{x^T Ax}{x^T x}$$

[2]Хорошая шпаргалка с разложением матриц доступна на сайте курса по линейной алгебре website.

и, следовательно, $\forall x \in \mathbb{R}^n$ (соотношение Рэлея):

$$\lambda_{\min}(A)x^Tx \le x^TAx \le \lambda_{\max}(A)x^Tx$$

Число обусловленности невырожденной матрицы определяется как

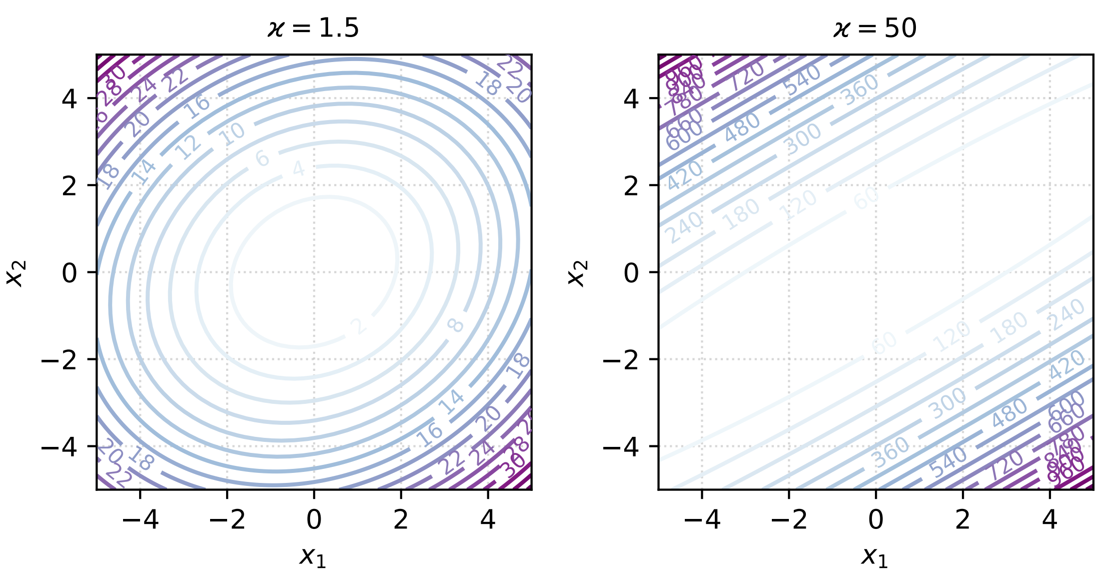
$$\kappa(A) = \|A\|\|A^{-1}\|$$

Если мы используем спектральную матричную норму, мы можем получить:

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

Если, кроме того, $A \in \mathbb{S}^n_{++}$: $\kappa(A) = \dfrac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$

1.13 Число обусловленности



1.14 Сингулярное разложение (SVD)

Пусть $A \in \mathbb{R}^{m\times n}$ с рангом $A = r$. Тогда A может быть разложена как

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

где $U \in \mathbb{R}^{m\times r}$ удовлетворяет $U^TU = I$, $V \in \mathbb{R}^{n\times r}$ удовлетворяет $V^TV = I$, и Σ является диагональной матрицей с $\Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_r)$, такой что

$$\sigma_1 \ge \sigma_2 \ge ... \ge \sigma_r > 0.$$

Это разложение называется **сингулярным разложением (SVD)** матрицы A. Столбцы U называются левыми сингулярными векторами A, столбцы V называются правыми сингулярными векторами, и числа σ_i являются сингулярными значениями. Сингулярное разложение может быть записано как

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

где $u_i \in \mathbb{R}^m$ являются левыми сингулярными векторами, и $v_i \in \mathbb{R}^n$ являются правыми сингулярными векторами.

1.15 Сингулярное разложение

> **Question**
>
> Пусть $A \in \mathbb{S}^n_{++}$. Что мы можем сказать о связи между его собственными значениями и сингулярными значениями?

> **Question**
>
> Как сингулярные значения матрицы связаны с её собственными значениями, особенно для симметричной матрицы?

1.16 Ранговое разложение (Skeleton decomposition)

Простое, но очень интересное разложение - это ранговое разложение, которое может быть записано в двух формах:

$$
A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}
$$

Последнее выражение относится к забавному факту: вы можете случайным образом выбрать r линейно независимых столбцов матрицы и любые r линейно независимых строк матрицы и хранить только их с возможностью точно (!) восстановить всю матрицу.

Применения для рангового разложения:

- Сжатие модели, сжатие данных и ускорение вычислений в численном анализе: для матрицы ранга r с $r \ll n, m$ необходимо хранить $\mathcal{O}((n+m)r) \ll nm$ элементов.
- Извлечение признаков в машинном обучении
- Все приложения, где применяется SVD, так как ранговое разложение может быть преобразовано в форму усеченного SVD.

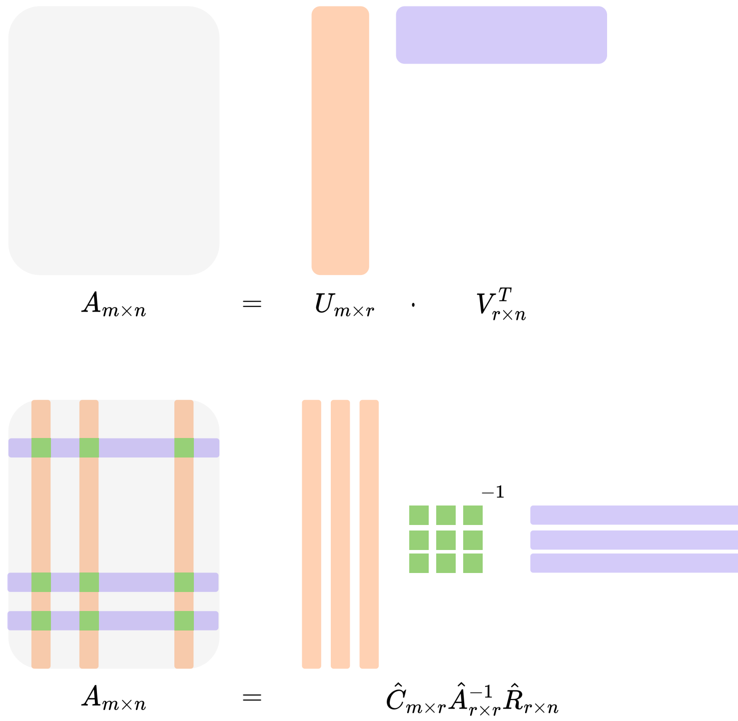


Рисунок 3: Иллюстрация рангового разложения

1.17 Каноническое тензорное разложение

Можно рассмотреть обобщение рангового разложения на структуры данных более высокого порядка, такие как тензоры, что означает представление тензора в виде суммы r простых тензоров.

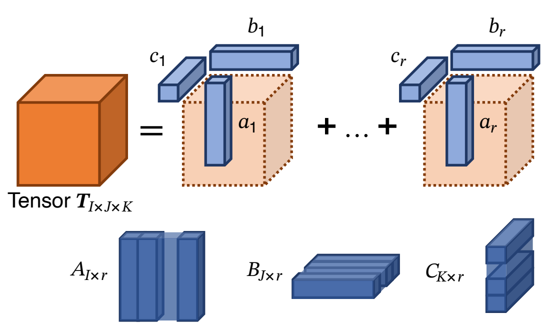


Рисунок 4: Иллюстрация канонического тензорного разложения

Example

Заметьте, что существует множество тензорных разложений: каноническое, Таккера, тензорный поезд (TT), тензорное кольцо (TR) и другие. В случае тензоров мы не имеем прямого определения *ранга* для всех типов разложений. Например, для разложения Тензорного поезда ранг является не скаляром, а вектором.

1.18 Определитель и след матрицы

Определитель и след матрицы могут быть выражены через собственные значения

$$\text{det}A = \prod_{i=1}^{n} \lambda_i, \qquad \text{tr}A = \sum_{i=1}^{n} \lambda_i$$

Определитель имеет несколько интересныхсвойств. Например,

- $\text{det}A = 0$ тогда и только тогда, когда A является вырожденной;
- $\text{det}AB = (\text{det}A)(\text{det}B)$;
- $\text{det}A^{-1} = \frac{1}{\det A}$.

Не забывайте о циклическом свойстве следа для произвольных матриц A, B, C, D (предполагая, что все размерности согласованы):

$$\text{tr}(ABCD) = \text{tr}(DABC) = \text{tr}(CDAB) = \text{tr}(BCDA)$$

> **Question**
>
> Как определитель матрицы связан с её обратимостью?

2 Скорости сходимости

2.1 Скорость сходимости

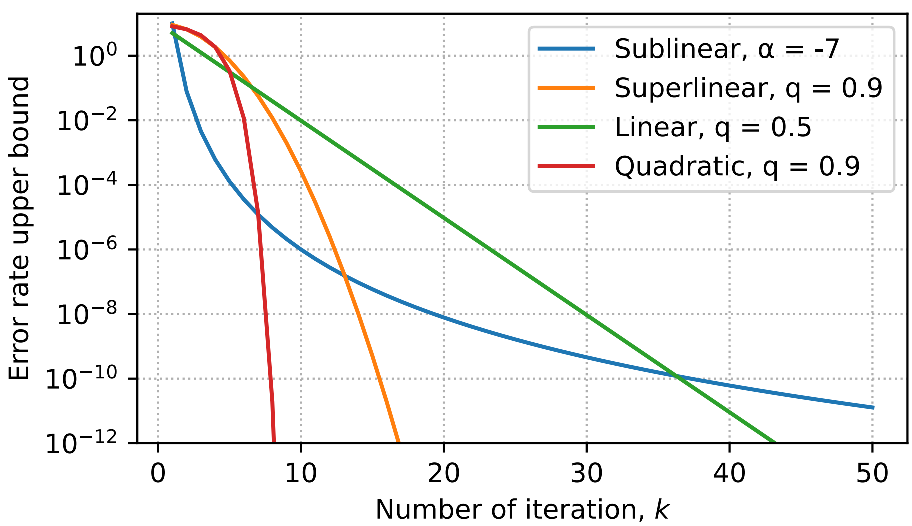


Рисунок 5: Разница в скоростях сходимости

2.2 *Линейная сходимость*

Чтобы сравнить производительность алгоритмов, мы должны определить термины для различных типов сходимости. Пусть r_k - последовательность неотрицательных вещественных чисел, которая сходится к нулю. Обычно мы имеем итерационный метод, который производит последовательность итераций x_k, приближающихся к оптимальному решению x^*, и $r_k = \|x_k - x^*\|_2$.

Линейная сходимость последовательности r_k определяется следующим образом:

Последовательность $\{r_k\}_{k=m}^\infty$ сходится *линейно* с параметром $0 < q < 1$, если существует константа $C > 0$ такая, что:

$$r_k \leq Cq^k, \quad \text{for all } k \geq m.$$

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

> **Question**
>
> Предположим, у вас есть две последовательности с линейными скоростями сходимости $q_1 = 0.1$ и $q_2 = 0.7$, какая из них быстрее?

2.3 *Линейная сходимость*

> **Example**
>
> Предположим, у нас есть следующая последовательность:
>
> $$r_k = \frac{1}{2^k}$$
>
> Можно сразу заключить, что мы имеем линейную сходимость с параметрами $q = \dfrac{1}{2}$ и $C = 0$.

> **Question**
>
> Определите сходимость следующей последовательности
>
> $$r_k = \frac{3}{2^k}$$

2.4 Сублинейная сходимость

Если последовательность r_k сходится к нулю, но не имеет *линейной* сходимости, то сходимость называется сублинейной. Иногда мы можем рассмотреть следующий частный случай сублинейной сходимости:

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

где $q < 0$ и $0 < C < \infty$. Интуитивно, сублинейная сходимость означает, что последовательность сходится медленнее любой геометрической прогрессии.

2.5 Сверхлинейная сходимость

Сходимость последовательности $\{r_k\}_{k=m}^\infty$ называется **сверхлинейной**, если она сходится к нулю быстрее любой линейно сходящейся последовательности. Проверьте, что последовательность $\{r_k\}_{k=m}^\infty$ является сверхлинейной, если она сходится линейно с параметром $q = 0$.

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

$$r_k \leq C q^{p^k}, \quad \text{for all } k \geq m.$$

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

> Важный пример
>
> Предположим, что $x^* = 1.23456789$ (истинное решение), и итерационная последовательность начинается с ошибки $r_k = 10^{-3}$, соответствующей 3 правильным значащим цифрам (1.234).
>
> 1. После первой итерации:
> $$r_{k+1} \approx r_k^2 = (10^{-3})^2 = 10^{-6}.$$
> Теперь ошибка равна 10^{-6}, и мы имеем 6 правильных значащих цифр (1.23456).
>
> 2. 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 правильных значащих цифр (1.234567890123).

2.6 Практические наблюдения о скоростях сходимости

- $\|x_{k+1} - x^*\|_2 \leq \frac{1}{k^{\frac{1}{p}}} \|x_0 - x^*\|_2$ означает сублинейную скорость сходимости
- $\|x_{k+1} - x^*\|_2 \leq q\|x_k - x^*\|_2$ означает линейную скорость сходимости, где $q < 1$
- $\|x_{k+1} - x^*\|_2 \leq q\|x_k - x^*\|_2^2$ означает квадратичную скорость сходимости, где $q\|x_0 - x^*\| < 1$

2.7 Тест корней

> Theorem
>
> Пусть $(r_k)_{k=m}^\infty$ - последовательность неотрицательных чисел, сходящаяся к нулю, и пусть $\alpha := \limsup_{k\to\infty} r_k^{1/k}$. (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.
>
> **Доказательство.**
>
> 1. Покажем, что если $(r_k)_{k=m}^\infty$ сходится линейно с константой $0 \leq \beta < 1$, то $\alpha \leq \beta$. Действительно, по определению константы линейной сходимости, для любого $\varepsilon > 0$ такого,

что $\beta + \varepsilon < 1$, существует $C > 0$ такое, что $r_k \leq C(\beta+\varepsilon)^k$ для всех $k \geq m$. Отсюда, $r_k^{1/k} \leq C^{1/k}(\beta+\varepsilon)$ для всех $k \geq m$. Переходя к пределу при $k \to \infty$ и используя $C^{1/k} \to 1$, мы получаем $\alpha \leq \beta + \varepsilon$. Учитывая произвольность ε, получаем $\alpha \leq \beta$.

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.

2.8 Тест корней

Theorem

1. 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 $k \geq N$. Отсюда, $r_k \leq (\alpha+\varepsilon)^k$ для всех $k \geq N$. Следовательно, $(r_k)_{k=m}^\infty$ сходится линейно с параметром $\alpha+\varepsilon$ (не имеет значения, что неравенство выполняется только для 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.

2. 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, которая не ограничена от нуля. Следовательно, $(r_k)_{k=m}^\infty$ не может сходиться к нулю, что противоречит условию.

2.9 Тест отношений

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

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

- Если существует q и $0 \leq q < 1$, то $\{r_k\}_{k=m}^\infty$ имеет линейную сходимость с константой q.
- В частности, если $q = 0$, то $\{r_k\}_{k=m}^\infty$ имеет сверхлинейную сходимость.
- Если q не существует, но $q = \lim\limits_{k\to\infty} \sup_k \dfrac{r_{k+1}}{r_k} < 1$, то $\{r_k\}_{k=m}^\infty$ имеет линейную сходимость с константой, не превышающей q.
- Если $\lim\limits_{k\to\infty} \inf_k \dfrac{r_{k+1}}{r_k} = 1$, то $\{r_k\}_{k=m}^\infty$ имеет сублинейную сходимость.
- Случай $\lim\limits_{k\to\infty} \inf_k \dfrac{r_{k+1}}{r_k} > 1$ невозможен.
- В остальных случаях (т.е., когда $\lim\limits_{k\to\infty} \inf_k \dfrac{r_{k+1}}{r_k} < 1 \leq \lim\limits_{k\to\infty} \sup_k \dfrac{r_{k+1}}{r_k}$) мы не можем сделать никаких конкретных утверждений о скорости сходимости $\{r_k\}_{k=m}^\infty$.

2.10 *Лемма о тесте отношений*

Theorem

Пусть $(r_k)_{k=m}^{\infty}$ - последовательность строго положительных чисел. (Строгая положительность необходима для того, чтобы отношения $\frac{r_{k+1}}{r_k}$, которые появляются ниже, были определены.) Тогда

$$\liminf_{k\to\infty} \frac{r_{k+1}}{r_k} \leq \liminf_{k\to\infty} r_k^{1/k} \leq \limsup_{k\to\infty} r_k^{1/k} \leq \limsup_{k\to\infty} \frac{r_{k+1}}{r_k}.$$

Доказательство.

1. Среднее неравенство следует из того, что liminf любой последовательности всегда меньше или равен её limsup. Докажем последнее неравенство; первое доказывается аналогично.

2. Обозначим $L := \limsup_{k\to\infty} \frac{r_{k+1}}{r_k}$. Если $L = +\infty$, то неравенство очевидно, поэтому предположим, что L конечно. Заметим, что $L \geq 0$, поскольку отношение $\frac{r_{k+1}}{r_k}$ положительно для всех $k \geq m$. Пусть $\varepsilon > 0$ - произвольное число. Согласно свойствам limsup, существует $N \geq m$ такое, что $\frac{r_{k+1}}{r_k} \leq L + \varepsilon$ для всех $k \geq N$. Отсюда, $r_{k+1} \leq (L+\varepsilon)r_k$ для всех $k \geq N$. Применяя индукцию, получаем $r_k \leq (L+\varepsilon)^{k-N} r_N$ для всех $k \geq N$. Пусть $C := (L+\varepsilon)^{-N} r_N$. Тогда $r_k \leq C(L+\varepsilon)^k$ для всех $k \geq N$, откуда $r_k^{1/k} \leq C^{1/k}(L+\varepsilon)$. Переходя к limsup при $k \to \infty$ и используя $C^{1/k} \to 1$, получаем $\limsup_{k\to\infty} r_k^{1/k} \leq L + \varepsilon$. Учитывая произвольность ε, получаем $\limsup_{k\to\infty} r_k^{1/k} \leq L$.

3 Задачи

3.1 Задача 1. Простая, но важная идея о матричных вычислениях.

Предположим, у вас есть следующее выражение

$$b = A_1 A_2 A_3 x,$$

где $A_1, A_2, A_3 \in \mathbb{R}^{3\times 3}$ - случайные квадратные плотные матрицы, и $x \in \mathbb{R}^n$ - вектор. Вам нужно вычислить b.

Какой способ лучше всего использовать?

1. $A_1 A_2 A_3 x$ (слева направо)
2. $(A_1 (A_2 (A_3 x)))$ (справа налево)
3. Не имеет значения
4. Результаты первых двух вариантов не будут одинаковыми.

Проверьте простой код после вашего интуитивного ответа.

3.2 Задача 2. Связь между Фробениусовой нормой и сингулярными значениями.

Пусть $A \in \mathbb{R}^{m \times n}$, и пусть $q := \min\{m, n\}$. Докажите, что

$$\|A\|_F^2 = \sum_{i=1}^{q} \sigma_i^2(A),$$

где $\sigma_1(A) \geq \ldots \geq \sigma_q(A) \geq 0$ - сингулярные значения матрицы A. Подсказка: используйте связь между Фробениусовой нормой и скалярным произведением и SVD.

3.3 Задача 3. Знайте свое скалярное произведение.

Упростите следующее выражение:

$$\sum_{i=1}^{n} \langle S^{-1} a_i, a_i \rangle,$$

где $S = \sum\limits_{i=1}^{n} a_i a_i^T, a_i \in \mathbb{R}^n, \det(S) \neq 0$

3.4 Задача 4. Простые скорости сходимости.

Определите сходимость (и её скорость) или расходимость следующих последовательностей:

- $r_k = \frac{1}{3^k}$
- $r_k = \frac{4}{3^k}$
- $r_k = \frac{1}{k^{10}}$
- $r_k = 0.707^k$
- $r_k = 0.707^{2^k}$

3.5 Задача 5. Один тест проще, чем другой.

Определите сходимость (и её скорость) или расходимость следующей последовательности:

$$r_k = \frac{1}{k^k}$$

3.6 Задача 6. Сверхлинейно, но не квадратично.

Покажите, что следующая последовательность не имеет квадратичной сходимости.

$$r_k = \frac{1}{3^{k^2}}$$

4 А где это нужно в реальной жизни?

4.1 LoRA: Low-Rank Adaptation of Large Language Models (arXiv:2106.09685)

Поскольку современные LLM слишком большие, чтобы вместиться в память среднего пользователя, мы используем некоторые трюки, чтобы сделать их потребление памяти меньше. Одним из наиболее популярных трюков является LoRA (Low-Rank Adaptation of Large Language Models).

Предположим, у нас есть матрица $W \in \mathbb{R}^{d \times k}$ и мы хотим выполнить следующее обновление:

$$W = W_0 + \Delta W.$$

Основная идея LoRA состоит в том, чтобы разложить обновление ΔW на две низкоранговые матрицы:

$$W = W_0 + \Delta W = W_0 + BA, \quad B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k},$$
$$rank(A) = rank(B) = r \ll \min\{d, k\}.$$

Проверьте ноутбук для примера реализации LoRA.

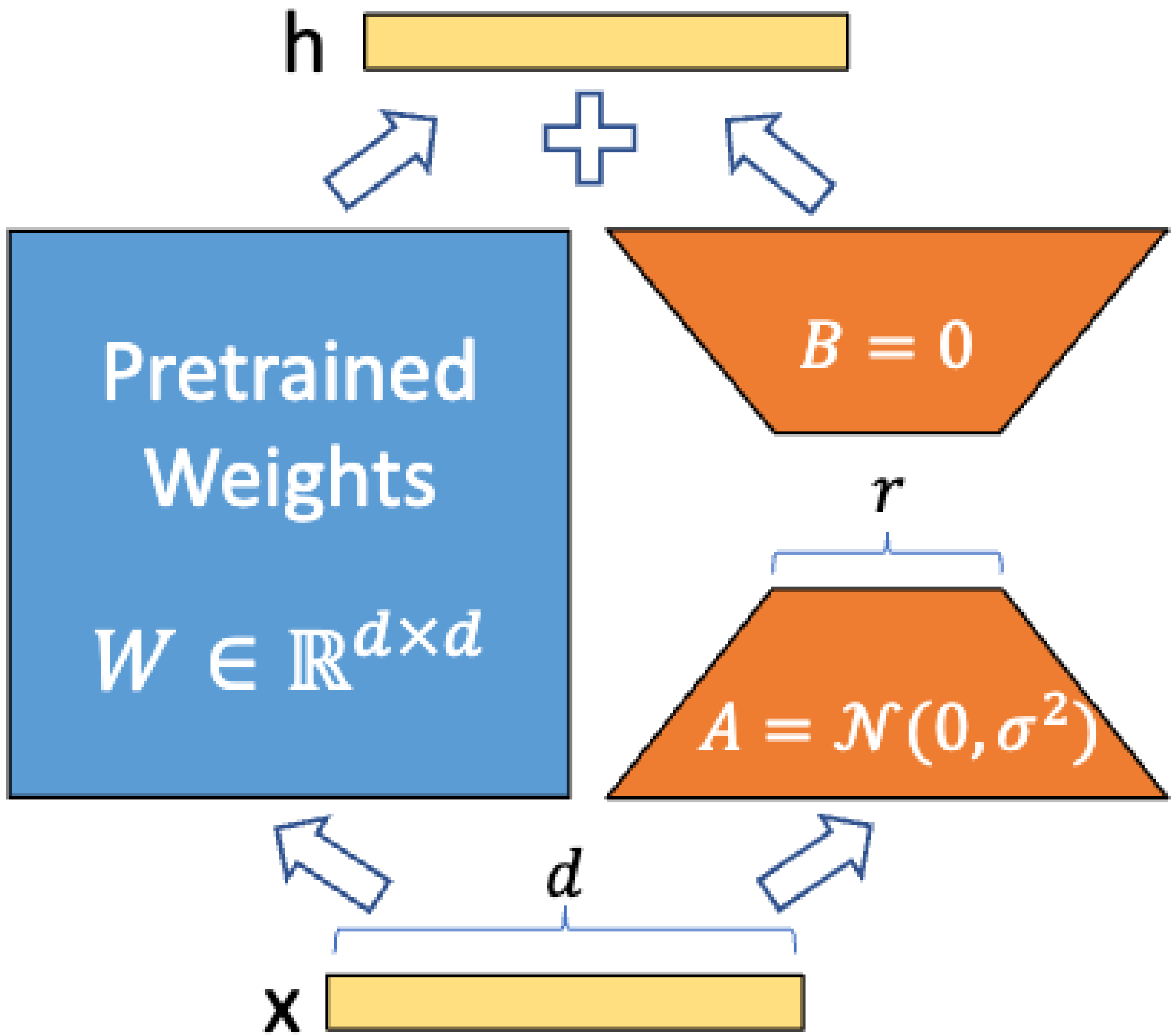


Рисунок 6: Иллюстрация LoRA

5 Задачи на дом

5.0.1 Вспоминаем линейную алгебру

1. [5 points] **Анализ чувствительности в линейных системах** Рассмотрим невырожденную матрицу $A \in \mathbb{R}^{n \times n}$ и вектор $b \in \mathbb{R}^n$. Предположим, что из-за ошибок измерения или вычислений вектор b изменяется на $\tilde{b} = b + \delta b$.

    1. Выведите верхнюю оценку относительной ошибки в решении x системы $Ax = b$ в терминах числа обусловленности $\kappa(A)$ и относительной ошибки в b.

2. Приведите конкретный пример использования матрицы 2×2, где $\kappa(A)$ велико (например, ≥ 100500).

2. [5 points] **Влияние диагонального масштабирования на ранг** Пусть $A \in \mathbb{R}^{n \times n}$ - матрица ранга r. Пусть $D \in \mathbb{R}^{n \times n}$ - диагональная матрица. Определите ранг произведения DA. Объясните ваше обоснование.

3. [8 points] **Неожиданный SVD** Вычислите сингулярное разложение (SVD) следующих матриц:

   - $A_1 = \begin{bmatrix} 2 \\ 2 \\ 8 \end{bmatrix}$
   - $A_2 = \begin{bmatrix} 0 & x \\ x & 0 \\ 0 & 0 \end{bmatrix}$, где x - сумма чисел вашего рождения (день + месяц).

4. [10 points] **Влияние нормализации на ранг** Предположим, у нас есть набор данных $x^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, m$, и мы решили представить эти данные в виде матрицы

$$X = \begin{pmatrix} | & & | \\ x^{(1)} & \ldots & x^{(m)} \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times m}.$$

Предположим, что $\operatorname{rank} X = r$.

В следующей задаче мы просим вас найти ранг некоторой матрицы M, связанной с X. В частности, вам нужно найти связь между $\operatorname{rank} X = r$ и $\operatorname{rank} M$, например, что ранг M всегда больше/меньше ранга X или что $\operatorname{rank} M = \operatorname{rank} X \big/ 35$. Аргументируйте ваш ответ и сделайте его как можно более точным.

Обратите внимание, что граничные случаи возможны в зависимости от структуры матрицы X. Убедитесь, что вы правильно освещаете их в своем ответе.

В прикладной статистике и машинном обучении данные часто нормализуются. Одна из наиболее популярных стратегий состоит в том, чтобы вычесть оцененное среднее μ и разделить на квадратный корень из оцененной дисперсии σ^2. т.е.

$$x \to (x - \mu) \big/ \sigma.$$

После нормализации мы получаем новую матрицу

$$\begin{split} Y &:= \begin{pmatrix} | & & | \\ y^{(1)} & \ldots & y^{(m)} \\ | & & | \end{pmatrix}, \\ y^{(i)} &:= \frac{x^{(i)} - \frac{1}{m}\sum_{j=1}^m x^{(j)}}{\sigma}. \end{split}$$

Каков ранг Y если $\operatorname{rank}\ X = r$? Здесь σ - вектор, и деление выполняется поэлементно. Причина этого в том, что разные признаки могут иметь разные масштабы. В частности:

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{j=1}^m \left(x_i^{(j)}\right)^2 - \left(\frac{1}{m}\sum_{j=1}^m x_i^{(j)}\right)^2}.$$

5. [20 points] **Сжатие изображений с использованием усеченного SVD** Исследуйте сжатие изображений с использованием усеченного сингулярного разложения (SVD). Понимание того, как изменение количества сингулярных значений влияет на качество сжатого изображения. Реализуйте Python скрипт для сжатия черно-белого изображения с использованием усеченного SVD и визуализируйте качество сжатия.

   - **Усеченное SVD**: Разлагает изображение A на матрицы U, S, и V. Сжатое изображение восстанавливается с использованием подмножества сингулярных значений.
   - **Математическое представление**:
   $$A \approx U_k \Sigma_k V_k^T$$
     - U_k и V_k - первые k столбцов U и V соответственно.
     - Σ_k - диагональная матрица с первыми k сингулярными значениями.
     - **Относительная ошибка**: Измеряет точность сжатого изображения по сравнению с оригиналом.
   $$\text{Relative Error} = \frac{\|A - A_k\|}{\|A\|}$$

```python
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import numpy as np
from skimage import io, color
import requests
from io import BytesIO

def download_image(url):
    response = requests.get(url)
    img = io.imread(BytesIO(response.content))
    return color.rgb2gray(img)  # Convert to grayscale

def update_plot(i, img_plot, error_plot, U, S, V, original_img, errors, ranks, ax1, ax2):
    # Adjust rank based on the frame index
    if i < 70:
        rank = i + 1
    else:
        rank = 70 + (i - 69) * 10

    reconstructed_img = ... # YOUR CODE HERE

    # Calculate relative error
    relative_error = ... # YOUR CODE HERE
    errors.append(relative_error)
    ranks.append(rank)

    # Update the image plot and title
    img_plot.set_data(reconstructed_img)
    ax1.set_title(f"Image compression with SVD\n Rank {rank}; Relative error {relative_error:.2

    # Remove axis ticks and labels from the first subplot (ax1)
    ax1.set_xticks([])
    ax1.set_yticks([])
```

```
    # Update the error plot
    error_plot.set_data(ranks, errors)
    ax2.set_xlim(1, len(S))
    ax2.grid(linestyle=":")
    ax2.set_ylim(1e-4, 0.5)
    ax2.set_ylabel('Relative Error')
    ax2.set_xlabel('Rank')
    ax2.set_title('Relative Error over Rank')
    ax2.semilogy()

    # Set xticks to show rank numbers
    ax2.set_xticks(range(1, len(S)+1, max(len(S)//10, 1)))  # Adjust the step size as needed
    plt.tight_layout()

    return img_plot, error_plot


def create_animation(image, filename='svd_animation.mp4'):
    U, S, V = np.linalg.svd(image, full_matrices=False)
    errors = []
    ranks = []

    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(5, 8))
    img_plot = ax1.imshow(image, cmap='gray', animated=True)
    error_plot, = ax2.plot([], [], 'r-', animated=True)  # Initial empty plot for errors

    # Add watermark
    ax1.text(1, 1.02, '@fminxyz', transform=ax1.transAxes, color='gray', va='bottom', ha='right

    # Determine frames for the animation
    initial_frames = list(range(70))  # First 70 ranks
    subsequent_frames = list(range(70, len(S), 10))  # Every 10th rank after 70
    frames = initial_frames + subsequent_frames

    ani = animation.FuncAnimation(fig, update_plot, frames=len(frames), fargs=(img_plot, error_
    ani.save(filename, writer='ffmpeg', fps=8, dpi=300)

    # URL of the image
    url = ""

    # Download the image and create the animation
    image = download_image(url)
    create_animation(image)
```

5.0.2 Скорости сходимости

1. [6 points] Определите (это означает определить характер сходимости, если она сходится) сходимость или расходимость следующих последовательностей

- $r_k = \frac{1}{\sqrt{k+5}}$.
- $r_k = 0.101^k$.
- $r_k = 0.101^{2^k}$.

2. [8 points] Пусть последовательность $\{r_k\}$ определена следующим образом

$$r_{k+1} = \begin{cases} \frac{1}{2}\, r_k, & \text{если } k \text{ чётно}, \\ r_k^2, & \text{если } k \text{ нечётно}, \end{cases}$$

с начальным значением $0 < r_0 < 1$. Докажите, что $\{r_k\}$ сходится к 0 и проанализируйте её скорость сходимости. В вашем ответе определите, является ли общая сходимость линейной, сублинейной или квадратичной.

3. [6 points] Определите скорость сходимости следующей последовательности $\{r_k\}$ (линейная, сублинейная, сверхлинейная). В случае сверхлинейной сходимости определите, является ли сходимость квадратичной.

$$r_k = \frac{1}{k!}$$

4. [8 points] Рассмотрим рекуррентную последовательность, определенную следующим образом

$$r_{k+1} = \lambda\, r_k + (1 - \lambda)\, r_k^p, \quad k \geq 0,$$

где $\lambda \in [0, 1)$ и $p > 1$. Какие дополнительные условия на r_0 должны быть выполнены для того, чтобы последовательность сходилась? Покажите, что когда $\lambda > 0$ последовательность сходится к 0 с линейной скоростью (с асимптотической константой λ), и когда $\lambda = 0$ определите скорость сходимости в терминах p. В частности, для $p = 2$ определите, является ли сходимость квадратичной.