Mathematics for Data Science — Lesson 6
Operator and Spectral Norm Solutions

Exercise 1: (Sequences — pedagogical) Work with $(x_n) \subset \mathbb{R}^d$ under any norm $\|\cdot\|$.

a) Show directly from the ε–N definition that $a_n = \dfrac{(-1)^n}{n} \to 0$.

b) Let $v_n = \left(\frac{1}{n}, \frac{(-1)^n}{n}, 1 - \frac{1}{n}\right) \in \mathbb{R}^3$. Prove $v_n \to (0, 0, 1)$ by checking coordinate-wise limits.

Solution. Understanding the task: This exercise builds foundational skills in working with sequence convergence using the precise ε–N definition and understanding convergence in $\mathbb{R}^d$.

a) **Proving $a_n = \frac{(-1)^n}{n} \to 0$ using the ε–N definition:**
Strategy: Recall that a sequence (a_n) converges to L if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that $|a_n - L| < \varepsilon$ for all $n \geq N$. Our goal is to show that $|a_n - 0|$ can be made arbitrarily small by choosing n large enough.
Key observation: The alternating sign $(-1)^n$ doesn't affect the distance from zero because we take absolute values: $|(-1)^n/n| = 1/n$.
Proof: Let $\varepsilon > 0$ be arbitrary. We need to find N such that $|a_n - 0| < \varepsilon$ for all $n \geq N$.
Notice that

$$|a_n - 0| = \left|\frac{(-1)^n}{n}\right| = \frac{|(-1)^n|}{n} = \frac{1}{n}.$$

We want $\frac{1}{n} < \varepsilon$, which is equivalent to $n > \frac{1}{\varepsilon}$.
Therefore, choose any integer $N > \frac{1}{\varepsilon}$ (for example, $N = \lceil \frac{1}{\varepsilon} \rceil$). Then for all $n \geq N$:

$$|a_n - 0| = \frac{1}{n} \leq \frac{1}{N} < \varepsilon.$$

Since ε was arbitrary, this proves $(a_n) \to 0$.
Remark: The sequence oscillates between positive and negative values, but the magnitude decreases to zero. This illustrates that convergence depends only on distance, not direction.
Common mistake: Students sometimes think the alternating sign prevents convergence. However, the limit is about $|a_n - L|$, which eliminates the sign.

b) **Proving $v_n \to (0, 0, 1)$ in $\mathbb{R}^3$ via coordinate-wise limits:**
Strategy: In finite-dimensional spaces like $\mathbb{R}^3$, convergence with respect to any norm is equivalent to coordinate-wise convergence. This is a powerful result that simplifies many proofs.

Theoretical foundation: For vectors in $\mathbb{R}^d$, the following are equivalent:

- $v_n \to v$ under some norm $\|\cdot\|$
- $v_n \to v$ under any norm $\|\cdot\|$ (equivalence of norms in finite dimensions)
- Each coordinate converges: $(v_n)_i \to v_i$ for all $i = 1, \ldots, d$

Proof: We have $v_n = \left(\frac{1}{n}, \frac{(-1)^n}{n}, 1 - \frac{1}{n}\right)$ and we want to show $v_n \to (0, 0, 1)$.

We check each coordinate separately:

- **First coordinate:** $\frac{1}{n} \to 0$ as $n \to \infty$ (standard limit, or use part (a) without the alternating sign)
- **Second coordinate:** $\frac{(-1)^n}{n} \to 0$ as $n \to \infty$ (this is exactly what we proved in part (a)!)
- **Third coordinate:** $1 - \frac{1}{n} \to 1$ as $n \to \infty$ because $\frac{1}{n} \to 0$, so $1 - \frac{1}{n} \to 1 - 0 = 1$

Since all three coordinates converge to their respective limits, we conclude that

$$v_n \to (0, 0, 1) \text{ in } \mathbb{R}^3$$

under any norm $\|\cdot\|$ on $\mathbb{R}^3$.

Geometric interpretation: The sequence starts near the point $(1, 1, 0)$ when $n = 1$ and gradually approaches $(0, 0, 1)$ along a spiral path (due to the alternating second coordinate). The first two coordinates shrink to zero while the third coordinate approaches 1.

Data science connection: In machine learning, we often track convergence of parameter vectors during optimization. Coordinate-wise convergence is useful because we can monitor each parameter independently. For example, in gradient descent, we might check that each weight component stabilizes, which guarantees overall convergence in any norm.

Exercise 2: (Asymptotics and rates) Analyse the following growth comparisons.

a) Show that the harmonic series partial sums $H_n = \sum_{k=1}^{n} \frac{1}{k}$ satisfy $H_n = \Theta(\log n)$.

b) Prove $n^\alpha = o(\beta^n)$ for every $\alpha > 0$ and $\beta > 1$.

c) Consider the recursion $x_{k+1} = \rho x_k + \dfrac{c}{k+1}$ with $x_0 \in \mathbb{R}$, $0 < \rho < 1$, $c > 0$. Find the asymptotic rate at which x_k tends to 0.

Solution. Understanding the task: Asymptotic analysis is crucial for understanding algorithm complexity and convergence rates. This exercise develops skills in comparing growth rates and analyzing recursive sequences.

a) **Showing $H_n = \Theta(\log n)$ for harmonic series partial sums:**

Recall: $H_n = \sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ and $f(n) = \Theta(g(n))$ means there exist positive constants c_1, c_2, N_0 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq N_0$.

Strategy: We'll use integral comparison to establish both upper and lower bounds on H_n in terms of $\log n$.

Proof:

Upper bound: Compare the sum to an integral. Since $f(x) = \frac{1}{x}$ is decreasing, we have

$$H_n = 1 + \sum_{k=2}^{n} \frac{1}{k} \leq 1 + \int_1^n \frac{dx}{x} = 1 + \log n.$$

Lower bound: Using the same decreasing property:

$$H_n = \sum_{k=1}^{n} \frac{1}{k} \geq \int_1^{n+1} \frac{dx}{x} = \log(n+1).$$

Combining the bounds: We have shown

$$\log(n+1) \leq H_n \leq 1 + \log n.$$

For large n, $\log(n+1) \approx \log n$, so both bounds are proportional to $\log n$. More precisely, choosing $c_1 = \frac{1}{2}$ (valid for $n \geq 2$) and $c_2 = 2$ (valid for $n \geq 1$), we get

$$\frac{1}{2} \log n \leq H_n \leq 2 \log n \quad \text{for sufficiently large } n.$$

Therefore, $H_n = \Theta(\log n)$.

Data science connection: The harmonic series appears in learning rate schedules. A learning rate $\eta_k = \frac{c}{k}$ gives total learning $\sum_{k=1}^{n} \eta_k = cH_n \approx c \log n$, which grows slowly but unboundedly—important for convergence guarantees in stochastic gradient descent.

b) **Proving $n^\alpha = o(\beta^n)$ for $\alpha > 0$ and $\beta > 1$:**

Recall: $f(n) = o(g(n))$ means $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. This says f grows strictly slower than g.

Intuition: Exponential functions eventually dominate any polynomial, no matter how large the polynomial degree or how small the exponential base (as long as $\beta > 1$). This fundamental result underlies why exponential-time algorithms are impractical for large inputs.

Strategy: We'll evaluate the limit $\lim_{n\to\infty} \frac{n^\alpha}{\beta^n}$ by taking logarithms to transform the product/quotient into a sum/difference, which is easier to analyze.

Proof: Consider the ratio

$$\frac{n^\alpha}{\beta^n} = \frac{n^\alpha}{e^{n \log \beta}} = e^{\alpha \log n - n \log \beta}.$$

Taking logarithms:

$$\log\left(\frac{n^\alpha}{\beta^n}\right) = \alpha \log n - n \log \beta.$$

Since $\beta > 1$, we have $\log \beta > 0$. The term $n \log \beta$ grows linearly in n, while $\alpha \log n$ grows logarithmically. Therefore:

$$\log\left(\frac{n^\alpha}{\beta^n}\right) = \alpha \log n - n \log \beta \to -\infty \quad \text{as } n \to \infty.$$

Exponentiating both sides:

$$\frac{n^\alpha}{\beta^n} = e^{\alpha \log n - n \log \beta} \to e^{-\infty} = 0.$$

Hence $n^\alpha = o(\beta^n)$.

Alternative approach: L'Hôpital's rule can be applied $\lceil \alpha \rceil$ times to show the limit is 0, though the logarithmic approach is more elegant.

Practical implication: This explains why algorithms with exponential time complexity ($O(2^n)$, $O(3^n)$, etc.) quickly become infeasible compared to polynomial algorithms ($O(n)$, $O(n^2)$, $O(n^{100})$, etc.), even for moderate n.

c) **Analyzing the recursive sequence $x_{k+1} = \rho x_k + \frac{c}{k+1}$:**

Context: This type of recursion appears in optimization with momentum (the ρx_k term) and diminishing step sizes (the $\frac{c}{k+1}$ term).

Strategy: We'll unfold the recursion to express x_k explicitly, then analyze its behavior by identifying two competing components: geometric decay and harmonic accumulation.

Step 1: Unfolding the recursion

Starting from x_0 and repeatedly applying the recursion:

$$\begin{aligned} x_1 &= \rho x_0 + \frac{c}{1} \\ x_2 &= \rho x_1 + \frac{c}{2} = \rho^2 x_0 + \rho \frac{c}{1} + \frac{c}{2} \\ &\vdots \\ x_k &= \rho^k x_0 + c \sum_{j=0}^{k-1} \rho^{k-1-j} \frac{1}{j+1}. \end{aligned}$$

Reindexing with $\ell = k - 1 - j$ (so $j + 1 = k - \ell$):

$$x_k = \rho^k x_0 + c \sum_{\ell=0}^{k-1} \frac{\rho^\ell}{k-\ell}.$$

Step 2: Analyzing the two components

Geometric term: $\rho^k x_0 = \mathcal{O}(\rho^k)$ decays exponentially fast (linear convergence rate) since $0 < \rho < 1$.

Harmonic sum: The challenging part is $S_k = \sum_{\ell=0}^{k-1} \frac{\rho^\ell}{k-\ell}$. We need to show this behaves like $\Theta(1/k)$.

Step 3: Splitting the sum for upper bound

Split the sum at $\ell = \lfloor k/2 \rfloor$:

$$S_k = \underbrace{\sum_{\ell=0}^{\lfloor k/2 \rfloor} \frac{\rho^\ell}{k-\ell}}_{\text{small } \ell} + \underbrace{\sum_{\ell=\lfloor k/2 \rfloor + 1}^{k-1} \frac{\rho^\ell}{k-\ell}}_{\text{large } \ell}.$$

First sum (small ℓ): For $\ell \leq k/2$, we have $k - \ell \geq k/2$, so:

$$\sum_{\ell=0}^{\lfloor k/2 \rfloor} \frac{\rho^\ell}{k-\ell} \leq \frac{2}{k} \sum_{\ell=0}^{\lfloor k/2 \rfloor} \rho^\ell \leq \frac{2}{k} \sum_{\ell=0}^{\infty} \rho^\ell = \frac{2}{k(1-\rho)}.$$

Second sum (large ℓ): For $\ell > k/2$, we have $k - \ell < k/2$, so $\frac{1}{k-\ell} \leq \frac{2}{k}$:

$$\sum_{\ell=\lfloor k/2 \rfloor+1}^{k-1} \frac{\rho^\ell}{k-\ell} \leq \frac{2}{k} \sum_{\ell=\lfloor k/2 \rfloor+1}^{k-1} \rho^\ell \leq \frac{2}{k} \cdot \frac{\rho^{k/2}}{1-\rho} = O(\rho^{k/2}/k).$$

Combining: $S_k \leq \frac{2}{k(1-\rho)} + O(\rho^{k/2}/k)$, so $S_k = O(1/k)$.

Step 4: Lower bound

Taking just the first term ($\ell = 0$):

$$S_k \geq \frac{\rho^0}{k-0} = \frac{1}{k}.$$

Step 5: Conclusion

We have established:

$$\frac{c}{k} \leq cS_k \leq \frac{2c}{(1-\rho)k} + O(\rho^{k/2}/k).$$

Since $\rho^{k/2} \to 0$ exponentially fast, the dominant term is $\Theta(1/k)$. Therefore:

$$x_k = \rho^k x_0 + cS_k = \mathcal{O}(\rho^k) + \Theta(1/k) = \Theta(1/k).$$

Interpretation: After an initial transient period (the $\rho^k x_0$ term dies out exponentially), the sequence decays at the slower harmonic rate $\Theta(1/k)$. The geometric momentum term provides fast initial decay, but the persistent noise (the $c/(k+1)$ term) prevents faster than $\Theta(1/k)$ convergence.

Optimization connection: This mirrors gradient descent with momentum and diminishing step sizes. The momentum (ρ) helps early convergence, but the persistent gradient noise with step size $1/k$ determines the final convergence rate.

Exercise 3: (Pedagogical) Consider $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ acting on $(\mathbb{R}^2, \|\cdot\|)$.

a) Compute $\|A\|_1$ and $\|A\|_\infty$ by hand.

b) Compute $A^\top A$ and deduce $\|A\|_2$ from its eigenvalues.

c) Identify a unit vector v such that $\|Av\|_2 = \|A\|_2$ and describe Av.

Solution.

a) **Computing $\|A\|_1$ (maximum column sum):**
The 1-norm is defined as $\|A\|_1 = \max_j \sum_i |a_{ij}|$. We sum the absolute values in each column:

- Column 1: $|2| + |1| = 3$
- Column 2: $|1| + |2| = 3$

Therefore, $\|A\|_1 = \max(3,3) = 3$.

Computing $\|A\|_\infty$ (maximum row sum):

The ∞-norm is defined as $\|A\|_\infty = \max_i \sum_j |a_{ij}|$. We sum the absolute values in each row:

- Row 1: $|2| + |1| = 3$
- Row 2: $|1| + |2| = 3$

Therefore, $\|A\|_\infty = \max(3,3) = 3$.

b) **Computing the spectral norm $\|A\|_2$:**

The spectral norm is defined as $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)}$, the square root of the largest eigenvalue of $A^\top A$.

First, we compute $A^\top A$:

$$A^\top A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 2\cdot 2 + 1\cdot 1 & 2\cdot 1 + 1\cdot 2 \\ 1\cdot 2 + 2\cdot 1 & 1\cdot 1 + 2\cdot 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}.$$

To find the eigenvalues, we solve the characteristic equation $\det(A^\top A - \lambda I) = 0$:

$$\det \begin{bmatrix} 5-\lambda & 4 \\ 4 & 5-\lambda \end{bmatrix} = (5-\lambda)^2 - 16 = \lambda^2 - 10\lambda + 25 - 16 = \lambda^2 - 10\lambda + 9 = 0.$$

Factoring: $(\lambda - 9)(\lambda - 1) = 0$, so the eigenvalues are $\lambda_1 = 9$ and $\lambda_2 = 1$.

Therefore, $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)} = \sqrt{9} = 3$.

c) **Finding the maximizing unit vector:**

We need to find a unit vector v such that $\|Av\|_2 = \|A\|_2 = 3$. This occurs when v is an eigenvector of $A^\top A$ corresponding to the largest eigenvalue $\lambda = 9$.

Solving $(A^\top A - 9I)v = 0$:

$$\begin{bmatrix} 5-9 & 4 \\ 4 & 5-9 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -4 & 4 \\ 4 & -4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

From the first equation: $-4v_1 + 4v_2 = 0 \Rightarrow v_1 = v_2$. A normalized eigenvector is:

$$v = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Now we verify:

$$Av = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ 3 \end{bmatrix} = 3 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3v.$$

Thus $\|Av\|_2 = \|3v\|_2 = 3\|v\|_2 = 3 \cdot 1 = 3 = \|A\|_2$.

Geometric interpretation: The vector $Av = 3v$ shows that A stretches v by a factor of 3 in its own direction, achieving the maximum stretching factor (the spectral norm).

Exercise 4: (Pedagogical) Let A be linear on $(\mathbb{R}^n, \|\cdot\|)$ and define $\|A\|_{\text{op}} = \sup_{\|x\|=1} \|Ax\|$.

a) Prove $\sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \|Ax\|$.

b) Conclude that $\|A\|_{\text{op}}$ equals the largest radius of $A(\mathbb{S}^{n-1})$ (image of the unit sphere).

Solution. Understanding the task: This exercise establishes a fundamental property of operator norms: the supremum over the unit ball equals the supremum over the unit sphere. This simplifies many proofs and computations.

a) **Proving** $\sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \|Ax\|$**:**
Strategy: We need to show both directions: $\leq$ and $\geq$. The $\leq$ direction is trivial since the unit sphere is contained in the unit ball. For the $\geq$ direction, we'll use a scaling argument to relate points in the ball to points on the sphere.
Proof:
*Direction 1 (*sup *over sphere* $\leq$ sup *over ball):* This is immediate since $\mathbb{S}^{n-1} \subset \mathbb{B}^n$ (the sphere is contained in the ball), so:

$$\sup_{\|x\|=1} \|Ax\| \leq \sup_{\|x\|\leq 1} \|Ax\| .$$

*Direction 2 (*sup *over ball* $\leq$ sup *over sphere):* Let x be any vector with $\|x\| \leq 1$. We need to show that $\|Ax\|$ is bounded by the supremum over the sphere.
Case 1: If $x = 0$, then $\|Ax\| = \|A \cdot 0\| = \|0\| = 0 \leq \sup_{\|u\|=1} \|Au\|$.
Case 2: If $x \neq 0$, we can factor out its norm. Write

$$x = \|x\| \cdot u \quad \text{where} \quad u = \frac{x}{\|x\|}.$$

Note that $\|u\| = \frac{\|x\|}{\|x\|} = 1$, so u lies on the unit sphere.
By linearity of A:

$$\|Ax\| = \|A(\|x\| \cdot u)\| = \|x\| \cdot \|Au\| .$$

Since $\|x\| \leq 1$ and $\|Au\| \leq \sup_{\|v\|=1} \|Av\|$:

$$\|Ax\| = \|x\| \cdot \|Au\| \leq 1 \cdot \sup_{\|v\|=1} \|Av\| = \sup_{\|v\|=1} \|Av\| .$$

Taking the supremum over all x with $\|x\| \leq 1$:

$$\sup_{\|x\|\leq 1} \|Ax\| \leq \sup_{\|u\|=1} \|Au\| .$$

Conclusion: Combining both directions:

$$\sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \|Ax\| .$$

Intuition: Any vector in the unit ball can be written as a scaled version of a vector on the unit sphere (with scaling factor ≤ 1). Since the operator norm is homogeneous, scaling down the input scales down the output proportionally, so the maximum is achieved on the boundary (the sphere).
Common mistake: Students sometimes forget to handle the case $x = 0$ separately, or assume the result without proving the scaling argument rigorously.

b) **Geometric interpretation of $\|A\|_{\text{op}}$:**
Recall: From part (a), we know $\|A\|_{\text{op}} = \sup_{\|x\|=1} \|Ax\|$.
Understanding the image: Consider the image of the unit sphere under A:

$$A(\mathbb{S}^{n-1}) = \{Ax : \|x\| = 1\}.$$

Each point Ax in this image has a certain distance from the origin, namely $\|Ax\|$.
Interpretation: The operator norm $\|A\|_{\text{op}}$ is the **largest distance** from the origin to any point in $A(\mathbb{S}^{n-1})$. In other words, it's the radius of the smallest ball centered at the origin that contains $A(\mathbb{S}^{n-1})$.
Why the supremum is reached on the sphere: From part (a), we know that for any x in the unit ball (including interior points), $\|Ax\| \leq \sup_{\|u\|=1} \|Au\|$. This means:

- The image $A(\mathbb{B}^n)$ of the entire unit ball is contained in a ball of radius $\|A\|_{\text{op}}$.
- Every point Ax for x in the interior can be written as a scaled version of a point Au on the boundary: $Ax = \|x\| \cdot Au$ where $\|x\| < 1$.
- Therefore, all points in $A(\mathbb{B}^n)$ lie on line segments (radii) from the origin through points in $A(\mathbb{S}^{n-1})$.
- The maximum distance occurs at a point on $A(\mathbb{S}^{n-1})$, not in the interior.

Visualization: Imagine the unit sphere in the input space. The linear map A transforms it into an ellipsoid (in general). The operator norm is the length of the longest semi-axis of this ellipsoid—the maximum stretching factor.
Data science connection: In neural networks, the operator norm of weight matrices controls how signals amplify or attenuate through layers. Bounding these norms (e.g., through spectral normalization) helps stabilize training and provides generalization guarantees. The operator norm measures the maximum amplification factor across all possible input directions on the unit sphere.

Exercise 5: **(Intermediate)** For a normal matrix A (i.e., $AA^\top = A^\top A$), show that $\|A\|_2 = \max_i |\lambda_i(A)|$.

Solution. Understanding the task: For normal matrices (those commuting with their transpose), the spectral norm equals the maximum absolute eigenvalue. This is a powerful result connecting spectral properties to operator norms.
What is a normal matrix? A matrix A is normal if $AA^\top = A^\top A$. Important examples include:

- Symmetric matrices: $A = A^\top$ (real eigenvalues, orthogonal eigenvectors)
- Skew-symmetric matrices: $A = -A^\top$ (purely imaginary eigenvalues)
- Orthogonal/unitary matrices: $A^\top A = I$ (all eigenvalues have magnitude 1)

Strategy: We'll use the spectral theorem to diagonalize A with an orthogonal change of basis, then compute the spectral norm of the diagonal form.
Proof:

Step 1: Apply the spectral theorem

Since A is normal, the spectral theorem guarantees that A can be orthogonally diagonalized. That is, there exists an orthogonal matrix Q (meaning $Q^\top Q = I$) and a diagonal matrix Λ such that:

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

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ contains the eigenvalues of A (possibly complex, but normality ensures orthogonal diagonalizability).

Step 2: Compute the spectral norm

Recall that the spectral norm is defined as:

$$\|A\|_2 = \sup_{\|x\|=1} \|Ax\|_2 .$$

For any unit vector x with $\|x\|_2 = 1$, we have:

$$\begin{aligned} \|Ax\|_2 &= \left\|Q\Lambda Q^\top x\right\|_2 \\ &= \left\|\Lambda Q^\top x\right\|_2 \quad \text{(since } Q \text{ is orthogonal, it preserves norms)} \end{aligned}$$

Let $y = Q^\top x$. Since Q is orthogonal, $\|y\|_2 = \left\|Q^\top x\right\|_2 = \|x\|_2 = 1$, so y is also a unit vector.

Therefore:

$$\|Ax\|_2 = \|\Lambda y\|_2 .$$

Step 3: Compute the norm of a diagonal matrix

For a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ and a vector $y = (y_1, \ldots, y_n)$:

$$\Lambda y = (\lambda_1 y_1, \lambda_2 y_2, \ldots, \lambda_n y_n).$$

The 2-norm is:

$$\|\Lambda y\|_2^2 = \sum_{i=1}^{n} |\lambda_i y_i|^2 = \sum_{i=1}^{n} |\lambda_i|^2 |y_i|^2 \leq \max_i |\lambda_i|^2 \sum_{i=1}^{n} |y_i|^2 = \max_i |\lambda_i|^2.$$

The maximum is achieved when y is the standard basis vector e_j corresponding to $j = \arg\max_i |\lambda_i|$.

Step 4: Conclusion

Taking the supremum over all unit vectors x:

$$\|A\|_2 = \sup_{\|x\|=1} \|Ax\|_2 = \sup_{\|y\|=1} \|\Lambda y\|_2 = \max_i |\lambda_i|.$$

Key insight: The orthogonal transformation Q doesn't change norms (it's an isometry), so the spectral norm of A equals the spectral norm of its diagonal form Λ, which is simply $\max_i |\lambda_i|$.

Why normality matters: General (non-normal) matrices cannot be orthogonally diagonalized. For such matrices, the spectral norm is determined by singular values, not eigenvalues: $\|A\|_2 = \sigma_1(A)$ (largest singular value), which can differ from $\max_i |\lambda_i(A)|$.

Data science connection: In PCA, the covariance matrix is symmetric (hence normal), so its spectral norm equals its largest eigenvalue. This largest eigenvalue represents the direction of maximum variance in the data. Knowing $\|\Sigma\|_2 = \lambda_{\max}$ helps bound condition numbers and understand numerical stability.

Exercise 6: (Intermediate) Let $A = U\Sigma V^\top$ be the SVD with $\sigma_1 \geq \cdots \geq \sigma_r > 0$.

a) Show that the best rank-k approximation in spectral norm is $A_k = \sum_{i=1}^k \sigma_i u_i v_i^\top$ and that $\|A - A_k\|_2 = \sigma_{k+1}$.

b) Show that A_k is also optimal for the Frobenius norm and that $\|A - A_k\|_F^2 = \sum_{i>k} \sigma_i^2$.

Solution. Understanding the task: This exercise proves one of the most important results in numerical linear algebra and data science: SVD truncation provides the best low-rank approximation in both spectral and Frobenius norms. This is the foundation of techniques like PCA, recommender systems, and image compression.

a) **Best rank-k approximation in spectral norm (Eckart–Young–Mirsky theorem):**
Recall the SVD: Any matrix $A \in \mathbb{R}^{m\times n}$ can be written as:

$$A = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,$$

where $r = \operatorname{rank}(A)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are singular values, and u_i, v_i are the left/right singular vectors.
The rank-k truncation: Define

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top.$$

This keeps the k largest singular values and their corresponding singular vectors.
Strategy: We'll prove two things: (1) A_k is optimal, and (2) the error is σ_{k+1}.
Proof sketch of optimality:
Step 1: Express the error. We have:

$$A - A_k = \sum_{i=k+1}^{r} \sigma_i u_i v_i^\top.$$

Step 2: Compute the spectral norm. The spectral norm of a sum of orthogonal rank-1 matrices (the $u_i v_i^\top$ are mutually orthogonal) is the maximum singular value:

$$\|A - A_k\|_2 = \max_{i>k} \sigma_i = \sigma_{k+1}.$$

Step 3: Show optimality. For any matrix B with $\text{rank}(B) \leq k$, we need to show $\|A - B\|_2 \geq \sigma_{k+1}$.

The null space of B has dimension at least $n - k$. The span of $\{v_1, \ldots, v_{k+1}\}$ has dimension $k+1$. By the pigeonhole principle, there exists a unit vector x in both spaces:

$$x \in \text{span}\{v_1, \ldots, v_{k+1}\} \cap \text{null}(B), \quad \|x\|_2 = 1.$$

Then $Bx = 0$, and we can write $x = \sum_{i=1}^{k+1} c_i v_i$ with $\sum c_i^2 = 1$. Thus:

$$\|(A - B)x\|_2 = \|Ax\|_2 = \left\| \sum_{i=1}^{k+1} c_i \sigma_i u_i \right\|_2$$
$$= \sqrt{\sum_{i=1}^{k+1} c_i^2 \sigma_i^2} \geq \sigma_{k+1} \sqrt{\sum_{i=1}^{k+1} c_i^2} = \sigma_{k+1}.$$

Since this holds for some unit vector x, we have $\|A - B\|_2 \geq \sigma_{k+1}$.

Therefore, A_k is optimal with error exactly σ_{k+1}.

Geometric intuition: The SVD decomposes A into orthogonal directions (singular vectors) with associated strengths (singular values). Truncation keeps the strongest k directions. The error is determined by the first discarded direction (σ_{k+1}).

b) **Best rank-k approximation in Frobenius norm:**

Recall: The Frobenius norm is $\|A\|_F^2 = \sum_{i,j} a_{ij}^2 = \text{trace}(A^\top A) = \sum_{i=1}^r \sigma_i^2$.

Strategy: The Frobenius norm of a sum of orthogonal terms equals the sum of their squared Frobenius norms (Pythagorean theorem).

Proof:

The error is:

$$A - A_k = \sum_{i=k+1}^{r} \sigma_i u_i v_i^\top.$$

Since the singular vectors are orthonormal, the rank-1 matrices $u_i v_i^\top$ are mutually orthogonal in the Frobenius inner product: $\langle u_i v_i^\top, u_j v_j^\top \rangle_F = 0$ for $i \neq j$.

By the Pythagorean theorem:

$$\|A - A_k\|_F^2 = \sum_{i=k+1}^{r} \left\| \sigma_i u_i v_i^\top \right\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2 \left\| u_i v_i^\top \right\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2.$$

(Note: $\left\| u_i v_i^\top \right\|_F = 1$ since u_i and v_i are unit vectors.)

Optimality: A similar argument using the pigeonhole principle shows that for any B with rank at most k:

$$\|A - B\|_F^2 \geq \sum_{i=k+1}^{r} \sigma_i^2,$$

with equality for $B = A_k$.

Relationship between norms: Note that $\|A - A_k\|_2 = \sigma_{k+1}$ is just the largest term in the sum $\|A - A_k\|_F^2 = \sum_{i>k} \sigma_i^2$. The Frobenius norm accounts for all discarded singular values, while the spectral norm only considers the largest.

Data science applications:

- **PCA:** The first k principal components give the best rank-k approximation of the data covariance/correlation matrix
- **Recommender systems:** Matrix factorization techniques use low-rank approximations to predict missing entries
- **Image compression:** Store only the top k singular values/vectors, achieving compression ratio proportional to $k/(m+n)$
- **Noise reduction:** Small singular values often represent noise; truncation can denoise data

Choosing k: In practice, examine the singular value spectrum (scree plot). Look for an "elbow" where σ_i drops sharply, or use cross-validation to balance approximation quality vs. model complexity.

Exercise 7: (Low-rank linear regression) Let $X \in \mathbb{R}^{n\times d}$, $Y \in \mathbb{R}^{n\times m}$, and fix $k \le \min(d,m)$. Consider

$$\min_{\operatorname{rank}(W)\le k} \|XW - Y\|_F \,. \tag{$\star$}$$

Assume X has full column rank (if not, replace inverses by pseudoinverses below).

a) Given the SVD $X = P\Sigma Q^\top$ with $\Sigma \in \mathbb{R}^{d\times d}$ invertible, set $Z = Q^\top W$. Show that $(\star)$ is equivalent to $\min_{\operatorname{rank}(Z)\le k} \|\Sigma Z - P^\top Y\|_F$.

b) Prove that an optimal solution is $Z^* = \Sigma^{-1}\,[\,P^\top Y\,]_k$, where $[\,\cdot\,]_k$ denotes the best rank-k approximation (by SVD truncation).

c) Conclude that $W^* = QZ^*$ solves $(\star)$. Comment on how the singular values of $P^\top Y$ influence the error.

Solution. Understanding the task: This exercise solves a constrained multi-output regression problem where we seek a low-rank weight matrix. This appears in multi-task learning, transfer learning, and reduced-rank regression, where we want to share structure across multiple output variables.

Problem setup: We want to minimize $\|XW - Y\|_F$ subject to $\operatorname{rank}(W) \le k$. The constraint encourages W to have low-rank structure, which can:

- Reduce overfitting by limiting model capacity
- Capture shared structure across tasks
- Enable interpretability through latent factor models

a) **Transforming to an equivalent problem:**
Strategy: Use the SVD of X to transform coordinates, simplifying the problem while preserving the objective value and rank constraints.
Proof:
Given $X = P\Sigma Q^\top$ where:
- $P \in \mathbb{R}^{n\times d}$ has orthonormal columns ($P^\top P = I_d$)
- $\Sigma \in \mathbb{R}^{d\times d}$ is diagonal and invertible (since X has full column rank)
- $Q \in \mathbb{R}^{d\times d}$ is orthogonal ($Q^\top Q = QQ^\top = I_d$)

Set $Z = Q^\top W$. This is a change of variables in the weight space. Note that:

- Since Q is orthogonal (invertible), $W = QZ$, so the mapping $W \leftrightarrow Z$ is bijective
- $\text{rank}(W) = \text{rank}(QZ) = \text{rank}(Z)$ (orthogonal maps preserve rank)

Now transform the objective:

$$\begin{aligned}\|XW - Y\|_F &= \left\|P\Sigma Q^\top W - Y\right\|_F \\ &= \left\|P(\Sigma Q^\top W) - Y\right\|_F \\ &= \|P(\Sigma Z) - Y\|_F \quad \text{(substituting } Z = Q^\top W\text{)}\end{aligned}$$

Since P has orthonormal columns, $P^\top P = I$, and we can use the orthogonal invariance of the Frobenius norm. Multiply both sides by $P^\top$:

$$\|P(\Sigma Z) - Y\|_F = \left\|P^\top(P\Sigma Z - Y)\right\|_F = \left\|\Sigma Z - P^\top Y\right\|_F.$$

The last equality uses the fact that $P^\top P = I$, so $P^\top P\Sigma Z = \Sigma Z$.

Therefore:

$$\min_{\text{rank}(W)\le k} \|XW - Y\|_F = \min_{\text{rank}(Z)\le k} \left\|\Sigma Z - P^\top Y\right\|_F.$$

Intuition: The SVD rotates and rescales the problem into "principal coordinates" where the optimization becomes simpler.

b) **Solving the transformed problem:**

Strategy: The transformed problem has a special structure: Σ is diagonal and invertible. We'll show the solution involves SVD truncation of $B = P^\top Y$.

Proof:

Let $B = P^\top Y \in \mathbb{R}^{d\times m}$. The problem is:

$$\min_{\text{rank}(Z)\le k} \|\Sigma Z - B\|_F.$$

Key insight: Since Σ is diagonal and invertible, we can "divide" by it. If there were no rank constraint, the unconstrained solution would be $Z = \Sigma^{-1}B$.

With the rank constraint $\text{rank}(Z) \le k$, we want to find the best rank-k matrix Z such that ΣZ approximates B.

Since Σ is diagonal with positive entries, multiplication by Σ scales each row independently. Writing $\Sigma = \text{diag}(s_1, \ldots, s_d)$:

$$(\Sigma Z)_{ij} = s_i Z_{ij}.$$

The Frobenius norm becomes:

$$\|\Sigma Z - B\|_F^2 = \sum_{i,j}(s_i Z_{ij} - B_{ij})^2.$$

To minimize this, we first find the best rank-k approximation to B:

$$[B]_k = \text{SVD truncation of } B \text{ to rank } k.$$

Then set:

$$Z^* = \Sigma^{-1}[B]_k.$$

Since Σ is diagonal with $\text{rank}(\Sigma) = d$, and $\text{rank}([B]_k) = k$:

$$\text{rank}(Z^*) = \text{rank}(\Sigma^{-1}[B]_k) = \text{rank}([B]_k) = k.$$

The error is:

$$\|\Sigma Z^* - B\|_F = \left\|\Sigma\Sigma^{-1}[B]_k - B\right\|_F = \|[B]_k - B\|_F .$$

By the Eckart-Young-Mirsky theorem (Exercise 6), this equals $\left(\sum_{i>k} \sigma_i(B)^2\right)^{1/2}$.

Why this works: Dividing by Σ "undoes" the scaling, allowing us to work directly with B. The optimal rank-k approximation to B is given by SVD truncation.

c) **Transforming back to the original variables:**

Strategy: Use $W = QZ$ to recover the solution in the original coordinates.

Solution:

Since $W = QZ$ and $Z^* = \Sigma^{-1}[P^\top Y]_k$:

$$W^* = QZ^* = Q\Sigma^{-1}[P^\top Y]_k.$$

This is the optimal low-rank weight matrix for the original problem.

Error analysis:

The residual error is:

$$\|XW^* - Y\|_F = \left\|\Sigma Z^* - P^\top Y\right\|_F = \|[B]_k - B\|_F = \left(\sum_{i>k} \sigma_i(B)^2\right)^{1/2},$$

where $\sigma_i(B)$ are the singular values of $B = P^\top Y$.

Interpretation: The error depends on the spectrum of $P^\top Y$:

- If $P^\top Y$ has rapidly decaying singular values, low-rank approximation works well
- The singular values $\sigma_i(P^\top Y)$ measure the intrinsic dimensionality of the relationship between X and Y
- If the first k singular values capture most of the "energy" ($\sum_{i=1}^k \sigma_i^2 \approx \sum_{i=1}^d \sigma_i^2$), then rank-$k$ approximation is accurate

Practical algorithm:

i. Compute SVD of X: $X = P\Sigma Q^\top$
ii. Compute $B = P^\top Y$
iii. Compute SVD of B and truncate to rank k: $[B]_k = \sum_{i=1}^k \sigma_i(B) u_i v_i^\top$
iv. Compute $W^* = Q\Sigma^{-1}[B]_k$

Data science applications:

- **Multi-task learning:** When predicting multiple related outputs (e.g., multiple languages in translation), low-rank W captures shared structure

- **Transfer learning:** Pre-trained features (rows of X) mapped through low-rank W to new tasks
- **Reduced-rank regression:** Classical statistics technique for high-dimensional multi-response regression
- **Collaborative filtering:** User-item matrices often have low intrinsic rank due to latent preferences

Choosing k: Use cross-validation or examine the singular value spectrum of $P^\top Y$ to balance model complexity and approximation quality.

Exercise 8: **(Pedagogical)** Determine convergence and find limits (when they exist) for the following sequences.

a) $a_n = \dfrac{3n^2 + 5n}{2n^2 - 1}$

b) $b_n = \dfrac{2^n + n^{10}}{3 \cdot 2^n - 5}$

c) $c_n = \sin(n\pi) + \dfrac{1}{n}$

d) $d_n = (-0.9)^n$

Solution. **Understanding the task:** This exercise develops skills in recognizing convergence patterns and computing limits using algebraic manipulation. These techniques are fundamental for analyzing algorithm behavior and convergence rates.

a) **Analyzing $a_n = \frac{3n^2+5n}{2n^2-1}$:**
Strategy: For rational functions (ratios of polynomials), compare the degrees of numerator and denominator. The limit behavior is determined by the highest-degree terms.
Proof:
Divide both numerator and denominator by n^2 (the highest power):

$$a_n = \frac{3n^2 + 5n}{2n^2 - 1} = \frac{n^2(3 + \frac{5}{n})}{n^2(2 - \frac{1}{n^2})} = \frac{3 + \frac{5}{n}}{2 - \frac{1}{n^2}}.$$

As $n \to \infty$:

- $\frac{5}{n} \to 0$
- $\frac{1}{n^2} \to 0$

Therefore, using limit arithmetic:

$$\lim_{n\to\infty} a_n = \frac{3 + 0}{2 - 0} = \frac{3}{2}.$$

General principle: For $a_n = \frac{p(n)}{q(n)}$ where p, q are polynomials:

- If $\deg(p) < \deg(q)$: $a_n \to 0$
- If $\deg(p) = \deg(q)$: $a_n \to \frac{\text{leading coeff of } p}{\text{leading coeff of } q}$
- If $\deg(p) > \deg(q)$: $a_n \to \pm\infty$ (diverges)

Here $\deg(p) = \deg(q) = 2$, so $a_n \to \frac{3}{2}$.

Geometric interpretation: As n grows large, lower-order terms ($5n$ in numerator, -1 in denominator) become negligible compared to n^2 terms. The sequence approaches the ratio of leading coefficients.

Data science connection: Many algorithm complexities are expressed as ratios of polynomials. For example, the average-case complexity of quicksort relative to merge sort can be analyzed using similar techniques. Understanding which terms dominate helps predict algorithm performance for large inputs.

Common mistake: Computing $\frac{3+5}{2-1} = 8$ by substituting $n = 1$. The limit is about behavior as $n \to \infty$, not at any finite point.

b) **Analyzing $b_n = \frac{2^n+n^{10}}{3\cdot 2^n-5}$:**

Strategy: When exponential and polynomial terms mix, exponentials dominate. Factor out the exponential term from both numerator and denominator.

Proof:

Factor out 2^n from numerator and denominator:

$$b_n = \frac{2^n + n^{10}}{3 \cdot 2^n - 5} = \frac{2^n(1 + \frac{n^{10}}{2^n})}{2^n(3 - \frac{5}{2^n})} = \frac{1 + \frac{n^{10}}{2^n}}{3 - \frac{5}{2^n}}.$$

We need to evaluate $\lim_{n\to\infty} \frac{n^{10}}{2^n}$. From Exercise 2b, we know that $n^\alpha = o(\beta^n)$ for any $\alpha > 0$ and $\beta > 1$. In particular, $\frac{n^{10}}{2^n} \to 0$.

Also, $\frac{5}{2^n} \to 0$ as $n \to \infty$ (exponential decay).

Therefore:

$$\lim_{n\to\infty} b_n = \frac{1+0}{3-0} = \frac{1}{3}.$$

Key insight: Exponential functions grow faster than any polynomial. Even though n^{10} grows rapidly, 2^n eventually dominates it completely. This is why we can factor out 2^n and the remaining term $\frac{n^{10}}{2^n}$ vanishes.

Geometric interpretation: Initially (for small n), n^{10} might be large compared to 2^n. But exponential growth is relentless—by $n = 50$, we have $2^{50} \approx 10^{15}$ while $50^{10} \approx 10^{17}$. The exponential catches up and eventually dominates.

Data science connection: Understanding exponential vs polynomial growth is critical for:

- **Algorithm complexity:** Exponential algorithms ($O(2^n)$) become infeasible quickly, while polynomial algorithms ($O(n^{10})$) remain manageable
- **Neural network width:** The number of possible configurations grows exponentially with layer width, dominating polynomial factors from depth
- **Regularization:** Exponential penalties (e.g., $e^{\lambda|w|}$) enforce sparsity more aggressively than polynomial penalties ($|w|^p$) for large weights

Common mistake: Thinking n^{10} dominates because "10 is a big exponent." Exponential functions with any base > 1 eventually dominate all polynomials, regardless of degree.

c) **Analyzing $c_n = \sin(n\pi) + \frac{1}{n}$:**

Strategy: Analyze each term separately, then use limit arithmetic (if both limits exist).

Proof:

First term: $\sin(n\pi)$ for integer n.

Since n is an integer:

$$\sin(n\pi) = \begin{cases} 0 & \text{if } n \text{ is any integer} \end{cases}$$

This is because $\sin(\pi) = 0$, $\sin(2\pi) = 0$, $\sin(3\pi) = 0$, etc. Therefore, $\sin(n\pi) = 0$ for all $n \in \mathbb{N}$.

Second term: $\frac{1}{n} \to 0$ as $n \to \infty$ (standard limit).

Combined:

$$c_n = \sin(n\pi) + \frac{1}{n} = 0 + \frac{1}{n} = \frac{1}{n} \to 0.$$

Therefore, $\lim_{n\to\infty} c_n = 0$.

Key insight: The periodic term $\sin(n\pi)$ evaluates to zero at all integer points, so it doesn't contribute to the limit. Only the decaying term $\frac{1}{n}$ matters.

Contrast with oscillatory sequences: If we had $c_n = \sin(n) + \frac{1}{n}$ (without the π), the sequence would oscillate and not converge, because $\sin(n)$ takes various values in $[-1, 1]$ without settling down. The π makes all the difference!

Geometric interpretation: Imagine plotting the sequence: all terms lie on the curve $y = \frac{1}{n}$ because the $\sin(n\pi)$ contribution is always zero. The sequence simply follows this decaying curve to zero.

Data science connection:

- **Cyclical patterns:** When analyzing time series with known periods (daily, weekly), we can exploit periodicity to simplify computations
- **Fourier analysis:** Understanding how periodic functions behave at specific sampling points is fundamental to signal processing
- **Step size schedules:** Some learning rate schedules include cyclical components (e.g., cosine annealing) that restart periodically, combined with an overall decay term

Common mistake: Thinking $\sin(n\pi)$ oscillates between -1 and 1 for integer n. It's crucial to evaluate the sine function at the specific points $n\pi$ where n is an integer.

d) **Analyzing $d_n = (-0.9)^n$:**

Strategy: This is a geometric sequence with ratio $r = -0.9$. Use the property that $|r| < 1$ implies $r^n \to 0$.

Proof:

Write $d_n = (-0.9)^n = (-1)^n \cdot (0.9)^n$.

The sequence alternates in sign:

- Odd n: $d_n = -(0.9)^n < 0$
- Even n: $d_n = (0.9)^n > 0$

However, the magnitude decreases:

$$|d_n| = |(-0.9)^n| = (0.9)^n \to 0 \quad \text{as } n \to \infty$$

since $0 < 0.9 < 1$ (geometric decay).

By the squeeze theorem (or directly): since $-|d_n| \leq d_n \leq |d_n|$ and both $\pm|d_n| \to 0$, we have $d_n \to 0$.

Therefore, $\lim_{n\to\infty} d_n = 0$.

General principle for geometric sequences: For $a_n = r^n$:

- If $|r| < 1$: $a_n \to 0$ (converges)
- If $r = 1$: $a_n = 1$ for all n (converges to 1)
- If $r = -1$: a_n oscillates between ± 1 (diverges)
- If $|r| > 1$: $|a_n| \to \infty$ (diverges)

Here $|r| = 0.9 < 1$, so convergence to 0 is guaranteed.

Geometric interpretation: The sequence oscillates like a damped wave: positive, negative, positive, negative, but with decreasing amplitude. Each term is 90% of the previous term's magnitude (with sign flip), so the oscillations decay exponentially to zero.

Convergence rate: This is *linear convergence* (also called exponential or geometric convergence): $|d_n| \leq C\rho^n$ with $\rho = 0.9$. After k iterations, the error is reduced by a factor of $\rho^k = (0.9)^k$. This is the same type of convergence as seen in gradient descent for strongly convex functions.

Data science connection:

- **Momentum in optimization:** Momentum methods use $m_t = \beta m_{t-1} + g_t$ where $\beta \approx 0.9$. The contribution of gradient g_0 after t steps decays as $\beta^t = (0.9)^t$
- **Exponential moving averages:** EMA in Adam optimizer uses similar decay: old gradients have exponentially decaying influence
- **Discount factors:** In reinforcement learning, future rewards are discounted by γ^t where $\gamma \in (0, 1)$, typically $\gamma \approx 0.9 - 0.99$
- **Convergence diagnostics:** If training loss decreases like $(-0.9)^n$, we have linear convergence with oscillations—acceptable but not optimal

Common mistakes:

- Thinking the sequence diverges because it changes sign: convergence only requires $|d_n - L| \to 0$
- Confusing geometric sequences ($a_n = r^n$) with arithmetic sequences ($a_n = a_0 + nd$)
- Not recognizing that $|r| < 1$ is the key condition for convergence

Summary of convergence patterns:

- **Rational functions:** Compare degrees of polynomials
- **Mixed exponential/polynomial:** Exponentials dominate
- **Periodic + decaying:** Evaluate periodic part at specific points
- **Geometric sequences:** Check if $|r| < 1$

These patterns appear repeatedly in algorithm analysis, making them essential tools for data scientists.

Exercise 9: (Intermediate) Work with sequences in $\mathbb{R}$ and $\mathbb{R}^d$.

a) Prove that every bounded monotone increasing sequence (a_n) in $\mathbb{R}$ converges.

b) Show that in finite-dimensional $\mathbb{R}^d$, a sequence is Cauchy if and only if it converges.

c) Given two norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ on $\mathbb{R}^d$, prove that if $(x_n) \to x$ coordinate-wise, then $(x_n) \to x$ in both norms.

Solution. Understanding the task: This exercise explores fundamental properties that distinguish finite-dimensional spaces from infinite-dimensional ones. These results are cornerstones of analysis and underpin convergence guarantees in optimization algorithms.

a) **Monotone Convergence Theorem for bounded sequences:**

Context: This theorem connects the order structure of $\mathbb{R}$ with its completeness. It's one of the most useful convergence tests because checking boundedness and monotonicity is often easier than guessing a limit.

Strategy: For an increasing bounded sequence, the limit should be the supremum (least upper bound). We'll use the completeness of $\mathbb{R}$ to guarantee this supremum exists, then show the sequence approaches it.

Proof:

Let (a_n) be a bounded increasing sequence. That is:

- *Increasing:* $a_n \leq a_{n+1}$ for all $n \in \mathbb{N}$
- *Bounded:* There exists $M \in \mathbb{R}$ such that $a_n \leq M$ for all n

Step 1: Define the limit candidate.

Since (a_n) is bounded above, the set $\{a_n : n \in \mathbb{N}\}$ has a supremum by the completeness of $\mathbb{R}$. Define:

$$L = \sup\{a_n : n \in \mathbb{N}\}.$$

Step 2: Show $a_n \to L$.

Let $\varepsilon > 0$ be arbitrary. By the definition of supremum:

- L is an upper bound: $a_n \leq L$ for all n
- $L - \varepsilon$ is not an upper bound: there exists N such that $a_N > L - \varepsilon$

Since the sequence is increasing, for all $n \geq N$:

$$L - \varepsilon < a_N \leq a_n \leq L < L + \varepsilon.$$

Therefore, $|a_n - L| < \varepsilon$ for all $n \geq N$, proving $a_n \to L$.

Geometric interpretation: The increasing sequence climbs toward its supremum like a staircase approaching a ceiling. Boundedness ensures the ceiling exists; monotonicity ensures the sequence can't retreat once it gets close.

Data science connection: Many iterative algorithms produce monotone decreasing objective sequences $f(x_k)$. If f is bounded below (e.g., $f \geq 0$ for loss functions), the Monotone Convergence Theorem guarantees the sequence of objective values converges. This doesn't guarantee x_k converges, but it does ensure the optimization process stabilizes. Examples include:

- Expectation-Maximization (EM) algorithm: likelihood is monotone increasing
- Coordinate descent: objective decreases at each step
- Proximal gradient methods: objective values form a decreasing bounded sequence

Common mistakes:

- *Forgetting boundedness:* The sequence $a_n = n$ is increasing but unbounded, and diverges to $+\infty$
- *Thinking monotonicity alone suffices:* Both conditions are necessary
- *Not using completeness:* This result fails in incomplete spaces like $\mathbb{Q}$ (rationals)

b) **Cauchy sequences and convergence in $\mathbb{R}^d$:**

Context: In finite dimensions, Cauchy sequences always converge—this is the completeness property. In infinite dimensions (e.g., function spaces), this can fail, which is why analysts carefully specify which spaces are complete.

Strategy: We'll prove both directions:

- ($\Rightarrow$) Convergent $\Rightarrow$ Cauchy: Use triangle inequality
- ($\Leftarrow$) Cauchy $\Rightarrow$ Convergent: Use coordinate-wise arguments and completeness of $\mathbb{R}$

Proof:

Direction 1: Convergent $\Rightarrow$ Cauchy.

Assume $(x_n) \to x^*$ in $(\mathbb{R}^d, \|\cdot\|)$. Let $\varepsilon > 0$. There exists N such that for all $n \geq N$:

$$\|x_n - x^*\| < \frac{\varepsilon}{2}.$$

For any $m, n \geq N$, by the triangle inequality:

$$\|x_n - x_m\| \leq \|x_n - x^*\| + \|x^* - x_m\| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Therefore, (x_n) is Cauchy.

Direction 2: Cauchy $\Rightarrow$ Convergent (finite dimensions).

Assume (x_n) is Cauchy in $\mathbb{R}^d$. Write $x_n = (x_n^{(1)}, \ldots, x_n^{(d)})$ in coordinates.

Step 1: Each coordinate sequence is Cauchy in $\mathbb{R}$.

For any norm $\|\cdot\|$ on $\mathbb{R}^d$, we have (by norm properties):

$$|x_n^{(i)} - x_m^{(i)}| \leq \|x_n - x_m\|$$

for each coordinate i. Since (x_n) is Cauchy, given $\varepsilon > 0$, there exists N such that $\|x_n - x_m\| < \varepsilon$ for all $m, n \geq N$. Therefore:

$$|x_n^{(i)} - x_m^{(i)}| < \varepsilon \quad \text{for all } m, n \geq N,$$

proving $(x_n^{(i)})$ is Cauchy in $\mathbb{R}$.

Step 2: Each coordinate sequence converges.

Since $\mathbb{R}$ is complete, each Cauchy sequence $(x_n^{(i)})$ converges to some limit $x^{*(i)} \in \mathbb{R}$. Define:

$$x^* = (x^{*(1)}, \ldots, x^{*(d)}).$$

Step 3: The vector sequence converges to x^*.

In finite dimensions, coordinate-wise convergence implies convergence in any norm (by norm equivalence, which we'll prove in part c). Therefore, $x_n \to x^*$ in $\mathbb{R}^d$.

Why finite dimension matters: The key step is coordinate-wise convergence $\Rightarrow$ convergence in the norm. This uses norm equivalence, which holds only in finite dimensions. In infinite dimensions (e.g., sequence spaces ℓ^2, function spaces L^2), Cauchy sequences may fail to converge, which is why we must carefully specify "complete" spaces (Banach spaces, Hilbert spaces).

Data science connection: In machine learning, parameter vectors $w \in \mathbb{R}^d$ live in finite-dimensional spaces, so:

- If gradient descent produces a Cauchy sequence of iterates, they converge to some point
- Convergence diagnostics can use the Cauchy criterion: if $\|w_k - w_{k-1}\|$ becomes small and stays small, we're near convergence
- In infinite-dimensional settings (kernel methods, neural networks with infinitely many parameters), Cauchy $\not\Rightarrow$ convergent without completeness assumptions

Common mistakes:

- *Assuming Cauchy $\Rightarrow$ convergent universally:* This fails in incomplete spaces
- *Forgetting to use completeness of $\mathbb{R}$:* The proof relies on each coordinate sequence converging
- *Circular reasoning:* Don't use "Cauchy $\Leftrightarrow$ convergent" when that's what you're proving

c) **Norm equivalence and coordinate-wise convergence:**

Context: All norms on $\mathbb{R}^d$ are equivalent, meaning they define the same topology and the same notion of convergence. This is a finite-dimensional phenomenon—in infinite dimensions, different norms can give different convergence behavior.

Strategy: We'll use the fact that all norms on finite-dimensional spaces are equivalent to coordinate-wise convergence (convergence in the ℓ^∞ norm). The key insight is that any norm is bounded above and below by the ℓ^∞ norm.

Proof:

Step 1: Coordinate-wise convergence means ℓ^∞ convergence.

If $x_n \to x$ coordinate-wise, then for each i, $x_n^{(i)} \to x^{(i)}$. This is equivalent to:

$$\|x_n - x\|_\infty = \max_{i=1,\ldots,d} |x_n^{(i)} - x^{(i)}| \to 0.$$

Step 2: Equivalence of any norm with ℓ^∞ norm.

For any norm $\|\cdot\|$ on $\mathbb{R}^d$, there exist constants $\alpha, \beta > 0$ such that:

$$\alpha \|z\|_\infty \leq \|z\| \leq \beta \|z\|_\infty \quad \text{for all } z \in \mathbb{R}^d.$$

This is a fundamental result in finite-dimensional analysis. The constants depend on the norm but not on the vector.

Step 3: Apply to convergence.

Given $\varepsilon > 0$, since $x_n \to x$ coordinate-wise, there exists N such that for all $n \geq N$:

$$\|x_n - x\|_\infty < \frac{\varepsilon}{\beta}.$$

Using the upper bound from norm equivalence:

$$\|x_n - x\| \leq \beta \|x_n - x\|_\infty < \beta \cdot \frac{\varepsilon}{\beta} = \varepsilon.$$

Therefore, $x_n \to x$ in the norm $\|\cdot\|$. The same argument works for any other norm.

Why this works: In finite dimensions, the unit sphere is compact, and all norms are continuous functions on this compact set. This forces them to be equivalent. In infinite dimensions, the unit sphere is not compact, and norms can behave very differently (e.g., ℓ^1 vs ℓ^2 vs ℓ^∞ on sequence spaces).

Geometric interpretation: Different norms correspond to different shapes of the unit ball (diamond for ℓ^1, sphere for ℓ^2, cube for ℓ^∞), but in finite dimensions, these shapes are all "roughly the same size" up to constant factors. A sequence escaping to infinity in one norm must escape in all norms; a sequence converging in one norm must converge in all norms.

Data science connection: In machine learning:

- **Convergence diagnostics:** Use any convenient norm (typically ℓ^2) to check convergence—the choice doesn't affect the conclusion
- **Regularization:** Different norms encourage different sparse structures (ℓ^1 promotes sparsity, ℓ^2 promotes smoothness), but convergence of optimization algorithms doesn't depend on which norm we monitor
- **Feature scaling:** Norm equivalence means convergence is robust to rescaling features (up to condition number effects)
- **High dimensions:** Even in very high dimensions (e.g., $d = 10^6$ in deep learning), as long as d is finite, norm equivalence holds

Common mistakes:

- *Confusing equivalence with equality:* Norms are equivalent (same convergence) but not equal (different values)
- *Applying to infinite dimensions:* In function spaces, different norms (e.g., L^1 vs L^2) give different notions of convergence
- *Forgetting constants:* The equivalence constants α, β can be large in high dimensions, affecting convergence rates in practice

Exercise 10: (Advanced) Explore deeper results about sequences and series.

a) State and prove the Monotone Convergence Theorem: every bounded monotone sequence in $\mathbb{R}$ converges. Explain how this extends to functions.

b) Use the integral test to show that $\sum_{k=1}^{\infty} \frac{1}{k^p}$ converges if and only if $p > 1$.

c) Prove the Bolzano–Weierstrass theorem: every bounded sequence in $\mathbb{R}$ has a convergent subsequence.

Solution. Understanding the task: These classical results form the foundation of real analysis and have direct applications to understanding convergence in optimization, regularization penalties, and clustering algorithms. They reveal the special structure of finite-dimensional spaces.

a) **Monotone Convergence Theorem (detailed proof):**

Context: This is one of the most fundamental convergence theorems in analysis. It characterizes when monotone sequences converge and provides a constructive way to find the limit (as the supremum or infimum).

Statement: Let (a_n) be a sequence in $\mathbb{R}$. If (a_n) is monotone increasing and bounded above, then (a_n) converges to $\sup_n a_n$. Similarly, if (a_n) is monotone decreasing and bounded below, then (a_n) converges to $\inf_n a_n$.

Strategy: We already proved the increasing case in Exercise 9a. Here we'll provide additional context and show how this extends to monotone functions.

Proof for increasing case: (See Exercise 9a for details)

Let $L = \sup_n a_n$ (exists by completeness of $\mathbb{R}$). For any $\varepsilon > 0$, there exists N with $a_N > L - \varepsilon$. Since (a_n) is increasing, $a_n \geq a_N > L - \varepsilon$ for all $n \geq N$. Also $a_n \leq L$ by definition of supremum. Therefore:

$$L - \varepsilon < a_n \leq L < L + \varepsilon \quad \text{for all } n \geq N,$$

proving $a_n \to L$.

Extension to monotone functions:

For a monotone increasing function $f : [a, b] \to \mathbb{R}$ that is bounded above, we can define:

$$L = \sup_{x \in [a,b]} f(x).$$

Then $f(x) \to L$ as $x \to b^-$ (left limit at the right endpoint). The proof is similar: given $\varepsilon > 0$, there exists $x_0 \in [a, b]$ with $f(x_0) > L - \varepsilon$. For all $x \in (x_0, b]$, monotonicity gives $f(x) \geq f(x_0) > L - \varepsilon$, while $f(x) \leq L$ by definition of supremum.

Interpretation: Monotone sequences/functions are "trapped" between their past values and their supremum/infimum. They can't oscillate or retreat, so if they're bounded, they must converge.

Data science applications:

- **EM algorithm:** The log-likelihood sequence is monotone increasing and bounded above (by the true maximum), guaranteeing convergence to a stationary point
- **Coordinate descent:** The objective function decreases monotonically; if bounded below, it converges
- **Learning rate schedules:** Diminishing step sizes $\eta_k = \frac{c}{k}$ satisfy $\sum \eta_k = \infty$ (needed for convergence) and $\sum \eta_k^2 < \infty$ (needed for variance control). The partial sums are monotone increasing and can be analyzed using this theorem

- **Boosting:** Training error decreases monotonically in AdaBoost; the Monotone Convergence Theorem guarantees it converges

Counterexample (unbounded): The sequence $a_n = n$ is monotone increasing but unbounded, and diverges to $+\infty$. Boundedness is essential.

Common mistakes:

- Forgetting that boundedness is necessary (monotone alone is insufficient)
- Confusing monotonicity of sequence with monotonicity of differences: $a_{n+1} - a_n \geq 0$ vs $a_{n+1} \geq a_n$
- Not recognizing when this theorem applies (many iterative algorithms produce monotone objective sequences)

b) **p-Series convergence via integral test:**

Context: The p-series $\sum_{k=1}^{\infty} \frac{1}{k^p}$ is fundamental in analysis and appears frequently in data science (regularization, learning rates, complexity analysis). The integral test provides a powerful way to determine convergence by comparing the sum to an integral.

Strategy: Compare the series to the integral $\int_1^{\infty} \frac{1}{x^p} dx$ using the monotonicity of $f(x) = \frac{1}{x^p}$. The series and integral have the same convergence behavior.

Proof:

Consider the function $f(x) = \frac{1}{x^p}$, which is positive, continuous, and decreasing on $[1, \infty)$ for $p > 0$.

Step 1: Upper bound (series $\leq$ integral).

For each $k \geq 1$, since f is decreasing:

$$\frac{1}{k^p} = f(k) \leq \int_{k-1}^{k} \frac{1}{x^p} dx.$$

Summing from $k = 2$ to n:

$$\sum_{k=2}^{n} \frac{1}{k^p} \leq \int_1^n \frac{1}{x^p} dx.$$

Step 2: Lower bound (integral $\leq$ series).

Similarly, since f is decreasing:

$$\int_k^{k+1} \frac{1}{x^p} dx \leq \frac{1}{k^p} = f(k).$$

Summing from $k = 1$ to $n - 1$:

$$\int_1^n \frac{1}{x^p} dx \leq \sum_{k=1}^{n-1} \frac{1}{k^p}.$$

Step 3: Analyze the integral.

Case 1: $p > 1$

$$\int_1^{\infty} \frac{1}{x^p} dx = \left[\frac{x^{-p+1}}{-p+1} \right]_1^{\infty} = \frac{1}{p-1} < \infty.$$

From the upper bound, $\sum_{k=2}^{n} \frac{1}{k^p} \leq \int_1^\infty \frac{1}{x^p} dx < \infty$, so the series converges.

Case 2: $p = 1$

$$\int_1^\infty \frac{1}{x} dx = [\log x]_1^\infty = \infty.$$

From the lower bound, $\sum_{k=1}^{n-1} \frac{1}{k} \geq \int_1^n \frac{1}{x} dx = \log n \to \infty$, so the series diverges.

Case 3: $p < 1$

$$\int_1^\infty \frac{1}{x^p} dx = \left[\frac{x^{-p+1}}{-p+1} \right]_1^\infty = \infty.$$

Similar to $p = 1$, the series diverges.

Conclusion: The series $\sum_{k=1}^{\infty} \frac{1}{k^p}$ converges if and only if $p > 1$.

Geometric interpretation: The series is comparing the area of rectangles (left/right Riemann sums) to the area under the curve $y = 1/x^p$. When $p > 1$, the curve decays fast enough that the total area is finite. When $p \leq 1$, the decay is too slow, and the area diverges.

Data science applications:

- **Regularization:** Ridge regression uses ℓ^2 penalty ($p = 2$); elastic net combines ℓ^1 and ℓ^2. Understanding p-series helps analyze the behavior of these penalties
- **Learning rates:** Step sizes $\eta_k = \frac{c}{k^p}$ give total learning $\sum \eta_k$, which diverges for $p \leq 1$ (needed for convergence) and converges for $p > 1$ (not enough learning). The variance $\sum \eta_k^2$ converges for $p > 1/2$, so common choices like $\eta_k = \frac{c}{k}$ ($p = 1$) balance these requirements
- **Complexity analysis:** Algorithms with $O(1/k^p)$ convergence rates have different behavior depending on p. For $p > 1$, the total error $\sum \frac{1}{k^p}$ is bounded; for $p \leq 1$, it grows unboundedly
- **Heavy-tailed distributions:** Power-law distributions with tail $P(X > x) \sim x^{-p}$ have finite mean when $p > 1$ and infinite mean otherwise

Connection to Riemann zeta function: For $p > 1$, the sum $\zeta(p) = \sum_{k=1}^{\infty} \frac{1}{k^p}$ defines the Riemann zeta function, which has deep connections to number theory and appears in statistical physics and machine learning (e.g., normalization constants in exponential families).

Common mistakes:

- Forgetting the boundary case $p = 1$ (harmonic series diverges)
- Mixing up $\sum \frac{1}{k}$ (diverges) with $\sum \frac{1}{k^2}$ (converges to $\pi^2/6$)
- Not recognizing when the integral test applies (need positive, continuous, decreasing)

c) **Bolzano–Weierstrass Theorem:**

Context: This theorem captures the compactness of bounded sets in $\mathbb{R}^d$. It guarantees that bounded sequences have convergent subsequences, which is crucial for existence proofs in optimization and for clustering algorithms.

Statement: Every bounded sequence in $\mathbb{R}$ has a convergent subsequence.

Strategy: We'll use a bisection argument (nested intervals) to construct a subsequence that converges. The key idea is to repeatedly halve intervals while ensuring infinitely many sequence elements remain in the chosen half.

Proof:

Let (a_n) be a bounded sequence in $\mathbb{R}$. Then there exist $M, m \in \mathbb{R}$ with $m \leq a_n \leq M$ for all n.

Step 1: Bisection construction.

Start with $I_0 = [m, M]$, which contains all terms of the sequence.

Iteration 1: Divide I_0 into two halves. At least one half contains infinitely many terms of (a_n). Choose such a half and call it $I_1 = [m_1, M_1]$.

Iteration 2: Divide I_1 into two halves. At least one half contains infinitely many terms of (a_n). Choose such a half and call it $I_2 = [m_2, M_2]$.

Continue this process to obtain a nested sequence of intervals:

$$I_0 \supseteq I_1 \supseteq I_2 \supseteq \cdots$$

where each $I_k = [m_k, M_k]$ contains infinitely many terms of (a_n), and the length of I_k is:

$$M_k - m_k = \frac{M - m}{2^k} \to 0.$$

Step 2: Construct the subsequence.

For each $k \geq 0$, choose an index n_k such that:

- $a_{n_k} \in I_k$
- $n_k > n_{k-1}$ (ensures (a_{n_k}) is a subsequence)

This is possible because each I_k contains infinitely many terms of (a_n).

Step 3: Show the subsequence is Cauchy.

For any j, k with $j \geq k$, both a_{n_j} and a_{n_k} lie in I_k (by nesting). Therefore:

$$|a_{n_j} - a_{n_k}| \leq M_k - m_k = \frac{M - m}{2^k}.$$

Given $\varepsilon > 0$, choose K such that $\frac{M-m}{2^K} < \varepsilon$. Then for all $j, k \geq K$:

$$|a_{n_j} - a_{n_k}| < \varepsilon,$$

proving (a_{n_k}) is Cauchy.

Step 4: The Cauchy subsequence converges.

Since $\mathbb{R}$ is complete, the Cauchy sequence (a_{n_k}) converges to some limit $L \in \mathbb{R}$.

Extension to $\mathbb{R}^d$: The same proof works in $\mathbb{R}^d$ by applying the bisection argument coordinate-wise or by using compactness of closed balls in finite dimensions.

Geometric interpretation: Bounded sequences can't escape to infinity, so they must have accumulation points (cluster points). The bisection process "corners" a subsequence around such a point.

Why boundedness matters: The unbounded sequence $a_n = n$ has no convergent subsequence (every subsequence diverges to $+\infty$).

Data science applications:

- **Optimization convergence:** If iterates (x_k) remain in a bounded set (e.g., constrained optimization), Bolzano-Weierstrass guarantees a convergent subsequence exists. This doesn't prove the whole sequence converges, but it guarantees limit points exist

- **Clustering algorithms:** k-means iterates stay bounded (centroids are weighted averages of data points). Even if the algorithm doesn't converge, there exists a convergent subsequence, providing a candidate clustering
- **Neural network training:** If gradients are clipped (keeping iterates bounded), Bolzano-Weierstrass guarantees subsequential convergence, though the full sequence may oscillate
- **Model selection:** In nested cross-validation, performance sequences are bounded. The theorem guarantees best-performing subsequences converge to optimal hyperparameters

Connection to compactness: Bolzano-Weierstrass is equivalent to sequential compactness of closed and bounded sets in $\mathbb{R}^d$. In infinite dimensions, this can fail: the unit ball in ℓ^2 (square-summable sequences) is closed and bounded but not sequentially compact.

Common mistakes:

- Thinking the whole sequence converges: B-W only guarantees a subsequence converges
- Forgetting boundedness: unbounded sequences may have no convergent subsequence
- Confusing limit points with limits: a sequence can have multiple limit points (via different subsequences) without converging

Exercise 11: (Limits and Continuity — Pedagogical) Work with limits and continuity in $\mathbb{R}$.

a) Prove using the ε–η definition that $\lim_{x\to 2}(3x+1) = 7$.

b) Use the squeeze theorem to show $\lim_{x\to 0}\dfrac{\sin x}{x} = 1$.

c) Compute the one-sided limits $\lim_{x\to 1^-} h(x)$ and $\lim_{x\to 1^+} h(x)$ for the piecewise function

$$h(x) = \begin{cases} x^2, & x < 1 \\ 2x, & x \geq 1. \end{cases}$$

Does $\lim_{x\to 1} h(x)$ exist? Is h continuous at $x = 1$?

d) Classify the discontinuity at $x = 1$ for $g(x) = \dfrac{x^2-1}{x-1}$ (removable) and at $x = 0$ for the step function $s(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$ (jump).

Solution. Understanding the task: This exercise builds foundational skills in rigorous limit definitions and continuity analysis. Mastering ε–η proofs develops precision in reasoning about function behavior, while understanding discontinuity types is crucial for analyzing loss landscapes and activation functions in machine learning.

a) **Proving $\lim_{x\to 2}(3x+1) = 7$ using ε–η definition:**

Strategy: Recall that $\lim_{x\to a} f(x) = \ell$ means: for every $\varepsilon > 0$, there exists $\eta > 0$ such that $|x - a| < \eta$ implies $|f(x) - \ell| < \varepsilon$. Our goal is to construct η from ε by working backwards from the desired inequality $|f(x) - \ell| < \varepsilon$.

Detailed proof:

Let $\varepsilon > 0$ be arbitrary. We need to find $\eta > 0$ such that

$$|x - 2| < \eta \Rightarrow |(3x + 1) - 7| < \varepsilon.$$

Step 1: Simplify the conclusion. We have

$$|(3x + 1) - 7| = |3x - 6| = |3(x - 2)| = 3\,|x - 2|\,.$$

Step 2: Connect to the hypothesis. We want $3\,|x - 2| < \varepsilon$, which is equivalent to

$$|x - 2| < \frac{\varepsilon}{3}.$$

Step 3: Choose η. Set $\eta = \frac{\varepsilon}{3}$. Then whenever $|x - 2| < \eta$, we have

$$|(3x + 1) - 7| = 3\,|x - 2| < 3\eta = 3 \cdot \frac{\varepsilon}{3} = \varepsilon.$$

Since ε was arbitrary, this proves $\lim_{x\to 2}(3x + 1) = 7$.

Interpretation and intuition: The ε–η definition formalizes "getting arbitrarily close." The factor of 3 in the function's slope means we need an η that is $1/3$ the size of ε to control the output. This inverse relationship between slope and required input tolerance is fundamental.

Geometric understanding: Draw horizontal bands at $y = 7 \pm \varepsilon$ and vertical bands at $x = 2 \pm \eta$. The proof shows that choosing $\eta = \varepsilon/3$ ensures the graph stays inside the horizontal band whenever we're inside the vertical band.

Data science connection: The ε–η framework underlies convergence guarantees in optimization. When analyzing gradient descent, we need to know how close to an optimum x^* we must be (the η) to guarantee the loss is within ε of the optimal value. The Lipschitz constant of the objective function (analogous to the slope 3 here) determines this relationship. Steeper functions require tighter tolerances.

Common mistakes:

- Setting $\eta = \varepsilon$ without accounting for the coefficient: this would give $|f(x) - 7| = 3\,|x - 2| < 3\varepsilon$, which is too large
- Trying to verify the limit at a specific point rather than for all x near 2
- Confusing the quantifier order: η depends on ε, not vice versa

b) **Using squeeze theorem for $\lim_{x\to 0} \frac{\sin x}{x} = 1$:**

Strategy: The squeeze (sandwich) theorem states: if $g(x) \le f(x) \le h(x)$ for x near a (except possibly at a) and $\lim_{x\to a} g(x) = \lim_{x\to a} h(x) = L$, then $\lim_{x\to a} f(x) = L$. We'll establish geometric inequalities that "trap" $\frac{\sin x}{x}$ between two functions that both approach 1.

Detailed proof:

Step 1: Geometric setup. Consider a unit circle. For $0 < x < \frac{\pi}{2}$, the arc of length x subtends an angle x (in radians). Draw the tangent line at the starting point of the arc.

Step 2: Area inequalities. The following areas satisfy:

$$\text{Area of triangle } OAB < \text{Area of sector } OAB < \text{Area of triangle } OAC,$$

where O is the origin, A is $(1, 0)$, B is $(\cos x, \sin x)$, and C is $(1, \tan x)$.

Computing these areas:

- Triangle OAB: $\frac{1}{2} \cdot 1 \cdot \sin x = \frac{\sin x}{2}$
- Sector OAB: $\frac{1}{2} \cdot 1^2 \cdot x = \frac{x}{2}$
- Triangle OAC: $\frac{1}{2} \cdot 1 \cdot \tan x = \frac{\tan x}{2}$

Step 3: Establish inequalities. From the area comparison:

$$\frac{\sin x}{2} < \frac{x}{2} < \frac{\tan x}{2}.$$

Multiplying by $\frac{2}{\sin x}$ (valid since $\sin x > 0$ for $0 < x < \frac{\pi}{2}$):

$$1 < \frac{x}{\sin x} < \frac{1}{\cos x}.$$

Taking reciprocals (and reversing inequalities):

$$\cos x < \frac{\sin x}{x} < 1.$$

Step 4: Apply limits. As $x \to 0^+$:

$$\lim_{x \to 0^+} \cos x = 1 \quad \text{and} \quad \lim_{x \to 0^+} 1 = 1.$$

By the squeeze theorem, $\lim_{x \to 0^+} \frac{\sin x}{x} = 1$.

Step 5: Handle negative x. For $x < 0$, note that $\frac{\sin x}{x} = \frac{\sin(-|x|)}{-|x|} = \frac{-\sin|x|}{-|x|} = \frac{\sin|x|}{|x|}$, so $\lim_{x \to 0^-} \frac{\sin x}{x} = 1$ by the same argument.

Therefore, $\lim_{x \to 0} \frac{\sin x}{x} = 1$.

Interpretation and intuition: The sine function and its argument are "approximately equal" for small x because $\sin x$ represents the vertical displacement along a unit circle, while x is the arc length. As the arc becomes smaller, the straight-line chord length (related to $\sin x$) approaches the arc length (x), making their ratio approach 1.

Geometric visualization: Imagine zooming in on a circle near $x = 0$. The arc, the chord, and the tangent line become indistinguishable, which is why $\sin x \approx x$ for small x. This is the first term in the Taylor series: $\sin x = x - \frac{x^3}{6} + \cdots$.

Data science connection:

- **Activation functions:** Many smooth activations (like tanh, swish) behave linearly near zero. Understanding $\lim_{x \to 0} \frac{\sin x}{x} = 1$ provides intuition for why neural networks can approximate linear functions in small neighborhoods—the derivative exists and is non-zero

- **Gradient analysis:** When analyzing gradient flow through layers with trigonometric activations, this limit tells us that small inputs don't vanish or explode (gradient is approximately 1)
- **Taylor approximations:** The limit $\frac{\sin x}{x} \to 1$ justifies using $\sin x \approx x$ in linearizations, which appears in small-angle approximations for physics-informed neural networks

Common mistakes:

- Substituting $x = 0$ directly to get $\frac{0}{0}$—this is indeterminate, not 0
- Forgetting to verify the limit from both sides (though symmetry helps here)
- Not recognizing this as a fundamental limit that cannot be computed by algebraic manipulation alone

c) **One-sided limits and continuity for piecewise function:**

Strategy: For piecewise functions, we must evaluate limits separately from each side using the appropriate piece of the definition. Continuity at a point requires three conditions: (1) the limit exists, (2) the function is defined there, and (3) the limit equals the function value.

Detailed solution:

Given $h(x) = \begin{cases} x^2, & x < 1 \\ 2x, & x \geq 1 \end{cases}$.

Step 1: Left-hand limit. As $x \to 1^-$ (from the left), we have $x < 1$, so we use $h(x) = x^2$:

$$\lim_{x \to 1^-} h(x) = \lim_{x \to 1^-} x^2 = 1^2 = 1.$$

Step 2: Right-hand limit. As $x \to 1^+$ (from the right), we have $x \geq 1$, so we use $h(x) = 2x$:

$$\lim_{x \to 1^+} h(x) = \lim_{x \to 1^+} 2x = 2(1) = 2.$$

Step 3: Two-sided limit. Since

$$\lim_{x \to 1^-} h(x) = 1 \neq 2 = \lim_{x \to 1^+} h(x),$$

the two-sided limit $\lim_{x \to 1} h(x)$ **does not exist**.

Step 4: Continuity. For continuity at $x = 1$, we need:

(i) $h(1)$ is defined: Yes, $h(1) = 2(1) = 2$

(ii) $\lim_{x \to 1} h(x)$ exists: No (by Step 3)

(iii) $\lim_{x \to 1} h(x) = h(1)$: Cannot hold since limit doesn't exist

Therefore, h is **not continuous** at $x = 1$.

Interpretation and intuition: The function "jumps" from value 1 to value 2 at $x = 1$. Approaching from the left, $h(x)$ tends toward 1 (following the parabola x^2), while approaching from the right, $h(x)$ follows the line $2x$ which starts at 2. This discontinuity gap of 1 cannot be bridged by any limiting process.

Geometric visualization: Graph $y = x^2$ for $x < 1$ (ending at the open circle $(1, 1)$) and $y = 2x$ for $x \geq 1$ (starting at the filled circle $(1, 2)$). The vertical gap between $(1, 1)$ and $(1, 2)$ visualizes the jump discontinuity.

Data science connection:

- **Activation functions:** ReLU has a "kink" (not a jump) at 0 where left and right derivatives differ. Understanding one-sided limits prepares us for analyzing such non-smooth activations
- **Loss landscapes:** Discontinuities in loss functions (e.g., from hard thresholding or quantization) create optimization challenges. Gradient-based methods struggle at jump discontinuities
- **Feature engineering:** Piecewise models (decision trees, spline regression) deliberately introduce discontinuities. Knowing where left and right limits differ helps interpret model boundaries
- **Regularization:** Jump discontinuities appear in proximal operators for ℓ^0 regularization. Subgradients generalize gradients to handle these cases

Common mistakes:

- Using the wrong piece: applying $h(x) = 2x$ when computing $\lim_{x\to 1^-}$ (must use x^2)
- Claiming the limit exists because $h(1)$ is defined—existence of $\lim_{x\to a} f(x)$ is independent of $f(a)$
- Confusing "limit exists but differs from $f(a)$" (removable) with "left and right limits differ" (jump)

d) **Classifying discontinuities:**

Strategy: A discontinuity at $x = a$ is:

- **Removable** if $\lim_{x\to a} f(x)$ exists but either $f(a)$ is undefined or $f(a) \neq \lim_{x\to a} f(x)$
- **Jump** if $\lim_{x\to a^-} f(x)$ and $\lim_{x\to a^+} f(x)$ both exist but are unequal
- **Essential/Infinite** if at least one one-sided limit fails to exist finitely

Analysis of $g(x) = \frac{x^2-1}{x-1}$ at $x = 1$:

Step 1: Check where g is defined. The denominator vanishes at $x = 1$, so $g(1)$ is undefined. However, for $x \neq 1$:

$$g(x) = \frac{x^2 - 1}{x - 1} = \frac{(x - 1)(x + 1)}{x - 1} = x + 1.$$

Step 2: Compute the limit. For $x \neq 1$:

$$\lim_{x\to 1} g(x) = \lim_{x\to 1}(x + 1) = 1 + 1 = 2.$$

Step 3: Classification. The limit exists ($= 2$) but $g(1)$ is undefined. This is a **removable discontinuity**. We can "fix" it by defining $\tilde{g}(x) = x + 1$ for all x, which is continuous everywhere.

Analysis of $s(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$ at $x = 0$:

Step 1: Compute one-sided limits.

$$\lim_{x\to 0^-} s(x) = \lim_{x\to 0^-}(-1) = -1, \quad \lim_{x\to 0^+} s(x) = \lim_{x\to 0^+} 1 = 1.$$

Step 2: Classification. Both one-sided limits exist but differ: $-1 \neq 1$. This is a **jump discontinuity** with jump magnitude $|1 - (-1)| = 2$.

Step 3: Cannot be removed. No redefinition of $s(0)$ can make s continuous at 0 because the left and right limits disagree.

Interpretation and intuition:

- **Removable:** The function has a "hole" that can be filled. Cancelling common factors reveals the "true" behavior
- **Jump:** The function makes a finite leap. The function is well-behaved on each side but incompatible at the junction

Data science connection:

- **Removable discontinuities:** Arise in numerical implementations with cancellation errors. For example, computing $\frac{\sin x}{x}$ directly at $x = 0$ fails, but the limit exists. Good libraries use Taylor series near 0 to "remove" the discontinuity
- **Jump discontinuities:**
  - Hard activation functions (step, sign) have jumps—gradients are zero almost everywhere, problematic for backprop
  - Quantization (discretizing weights/activations) introduces jumps—requires special handling (straight-through estimators)
  - Decision boundaries in classification can create jump discontinuities in prediction functions
- **Loss design:** Smooth losses (cross-entropy, MSE) avoid discontinuities for stable optimization. Huber loss transitions smoothly between regimes, avoiding jumps
- **Regularization paths:** As regularization strength varies, solutions may jump between different sparsity patterns (e.g., in LASSO)

Common mistakes:

- Thinking all discontinuities are removable—jumps cannot be fixed by redefining a single point
- Confusing "undefined at a point" with "discontinuous"—$g(x) = x + 1$ has no discontinuity at $x = 1$
- Not simplifying before evaluating limits—missing cancellations leads to incorrect "$\frac{0}{0}$" conclusions

Exercise 12: (Continuity Concepts — Pedagogical) Explore deeper continuity properties.

a) Determine whether each expression represents an indeterminate form or a definite answer:

(i) $\lim_{x\to 2} \dfrac{x^2 - 4}{x - 2}$

(ii) $\lim_{x\to 0} \dfrac{\sin x}{x^2}$

(iii) $\lim_{x\to 1} \dfrac{2}{x - 1}$

b) Prove that if (x_n) is a sequence with $x_n \to a$ and f is continuous at a, then $f(x_n) \to f(a)$ (sequential characterization of continuity).

c) Show by example that if $\lim_{x\to a^-} f(x) = \lim_{x\to a^+} f(x)$ does not guarantee f is continuous at a unless this common value also equals $f(a)$.

Solution. Understanding the task: This exercise develops conceptual understanding of indeterminate forms, sequential continuity, and the precise definition of continuity. These concepts underpin rigorous analysis of neural network behaviors, convergence proofs, and numerical stability.

a) **Indeterminate forms vs definite behavior:**

Strategy: An indeterminate form like "$\frac{0}{0}$" means we cannot determine the limit without further analysis (factoring, L'Hôpital, etc.). Forms like "$\frac{c}{0}$" with $c \neq 0$ indicate the function blows up (limit is $\pm\infty$ or does not exist).

Analysis:

(i) $\lim_{x\to 2} \frac{x^2-4}{x-2}$*:*

Direct substitution gives $\frac{2^2-4}{2-2} = \frac{0}{0}$ — **indeterminate form**.

To evaluate, factor the numerator:

$$\frac{x^2-4}{x-2} = \frac{(x-2)(x+2)}{x-2} = x+2 \quad \text{for } x \neq 2.$$

Therefore:

$$\lim_{x\to 2} \frac{x^2-4}{x-2} = \lim_{x\to 2}(x+2) = 4.$$

Interpretation: The "$\frac{0}{0}$" form signals that numerator and denominator share a common factor. Cancelling this factor reveals the true behavior. The indeterminacy arises from competing rates at which numerator and denominator approach zero.

(ii) $\lim_{x\to 0} \frac{\sin x}{x^2}$*:*

Direct substitution gives $\frac{\sin 0}{0^2} = \frac{0}{0}$ — **indeterminate form**.

Rewrite using $\frac{\sin x}{x}$:

$$\frac{\sin x}{x^2} = \frac{\sin x}{x} \cdot \frac{1}{x}.$$

We know $\lim_{x\to 0} \frac{\sin x}{x} = 1$ (Exercise 11b). Thus:

$$\lim_{x\to 0} \frac{\sin x}{x^2} = \lim_{x\to 0} \left(\frac{\sin x}{x} \cdot \frac{1}{x} \right) = 1 \cdot \lim_{x\to 0} \frac{1}{x}.$$

But $\lim_{x\to 0} \frac{1}{x}$ does not exist (approaches $+\infty$ from right, $-\infty$ from left). Therefore, **the limit does not exist**.

Interpretation: Even though we start with "$\frac{0}{0}$", resolving the indeterminacy reveals that the function blows up. The denominator dominates because x^2 goes to zero faster than $\sin x$ (which behaves like x).

(iii) $\lim_{x\to 1} \frac{2}{x-1}$*:*

Direct substitution gives $\frac{2}{1-1} = \frac{2}{0}$ — **NOT indeterminate**.

Since the numerator is nonzero and the denominator approaches 0:

- From the left ($x \to 1^-$): $x - 1 < 0$, so $\frac{2}{x-1} \to -\infty$

- From the right ($x \to 1^+$): $x - 1 > 0$, so $\frac{2}{x-1} \to +\infty$

Therefore, **the limit does not exist** (vertical asymptote at $x = 1$).

Interpretation: "$\frac{c}{0}$" with $c \neq 0$ indicates blow-up, not indeterminacy. The function grows without bound, with sign depending on the direction of approach.

Summary table:

Expression	Form	Behavior
(i)	$\frac{0}{0}$ (indeterminate)	Limit exists: $= 4$
(ii)	$\frac{0}{0}$ (indeterminate)	Limit DNE: $\to \pm\infty$
(iii)	$\frac{c}{0}$ (definite)	Limit DNE: vertical asymptote

Data science connection:

- **Numerical stability:** Computing "$\frac{0}{0}$" forms naively causes NaN errors. Recognizing indeterminacy prompts using Taylor series, L'Hôpital, or algebraic simplification
- **Gradient computation:** Second-order methods involve expressions like $\frac{\nabla^2 f}{\nabla f}$. Near critical points, both numerator and denominator vanish—proper handling prevents instability
- **Activation functions:** Sigmoid/tanh have $\frac{e^x - e^{-x}}{e^x + e^{-x}}$ form. For large $|x|$, naively computing creates overflow; recognizing limiting behavior guides stable implementations
- **Learning rate schedules:** Rates like $\frac{1}{t}$ or $\frac{1}{\sqrt{t}}$ approach infinity backward (at $t = 0$). Understanding "$\frac{c}{0}$" prevents divide-by-zero at initialization

Common mistakes:

- Concluding "$\frac{0}{0} = 0$" or "$\frac{c}{0} = \infty$" without proper analysis
- Thinking all "$\frac{0}{0}$" limits exist—(ii) shows they can diverge
- Not checking one-sided limits for "$\frac{c}{0}$" forms

b) **Sequential continuity theorem:**

Strategy: We'll use the ε–δ definition of continuity to construct an N that works for the sequence convergence. The key insight is that continuity provides a "tube" around $f(a)$, and sequence convergence ensures we eventually stay in the input region that maps into this tube.

Detailed proof:

Given:

- (x_n) is a sequence with $x_n \to a$ in $\mathbb{R}$
- $f : \mathbb{R} \to \mathbb{R}$ is continuous at a

To prove: $f(x_n) \to f(a)$.

Proof:

Let $\varepsilon > 0$ be arbitrary. We must find $N \in \mathbb{N}$ such that

$$n \geq N \Rightarrow |f(x_n) - f(a)| < \varepsilon.$$

Step 1: Use continuity of f at a. Since f is continuous at a, by the ε–δ definition, there exists $\delta > 0$ such that

$$|x - a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon.$$

Step 2: Use convergence of (x_n) *to* a. Since $x_n \to a$, there exists $N \in \mathbb{N}$ such that

$$n \geq N \Rightarrow |x_n - a| < \delta.$$

Step 3: Combine. For all $n \geq N$:

$$|x_n - a| < \delta \quad \overset{\text{Step 1}}{\Rightarrow} \quad |f(x_n) - f(a)| < \varepsilon.$$

Since ε was arbitrary, this proves $f(x_n) \to f(a)$.

Interpretation and intuition: Continuity means "small input changes lead to small output changes." If a sequence of inputs converges (inputs get close to a), then the sequence of outputs must converge (outputs get close to $f(a)$). The function cannot "break" convergence.

Geometric visualization: Draw horizontal bands at $y = f(a) \pm \varepsilon$ (output tolerance) and vertical bands at $x = a \pm \delta$ (input tolerance guaranteed by continuity). Since $x_n \to a$, the sequence eventually stays in the vertical band. Continuity ensures these points map into the horizontal band, so $f(x_n) \to f(a)$.

Converse (for completeness): If $f(x_n) \to f(a)$ for every sequence $x_n \to a$, then f is continuous at a. This gives an equivalent characterization: f is continuous at a iff it preserves convergence of sequences.

Data science connection:

- **Training stability:** If parameter updates w_t converge to w^*, continuity of the loss $\mathcal{L}$ ensures $\mathcal{L}(w_t) \to \mathcal{L}(w^*)$. Discontinuous losses break this guarantee, causing optimization instability
- **Prediction consistency:** If inputs x_n approach a test point x^* and the model f is continuous, predictions $f(x_n)$ must approach $f(x^*)$. This is essential for model reliability
- **Convergence diagnostics:** Monitor both parameter convergence ($w_t \to w^*$) and objective convergence ($\mathcal{L}(w_t) \to \mathcal{L}(w^*)$). Mismatch suggests discontinuity or numerical issues
- **Regularization:** Adding smooth regularizers (e.g., ℓ^2 penalty) preserves continuity, ensuring stable optimization. Non-smooth regularizers (ℓ^0, ℓ^1) require special handling

Common mistakes:

- Assuming the converse without proof—not every function that preserves some sequence convergences is continuous
- Forgetting to use the ε–δ definition of continuity—it's essential for connecting sequence and function convergence
- Thinking continuity is automatic—many ML objectives (with hard constraints, quantization) are discontinuous

c) **Continuity requires matching function value:**

Strategy: We'll construct a counterexample where left and right limits agree but differ from the function value at that point. This shows that having matching one-sided limits is necessary but not sufficient for continuity.

Counterexample:

Define

$$f(x) = \begin{cases} x^2, & x \neq 1 \\ 5, & x = 1. \end{cases}$$

Step 1: Compute one-sided limits at $x = 1$.
As $x \to 1$ from either side (with $x \neq 1$), we use $f(x) = x^2$:

$$\lim_{x \to 1^-} f(x) = \lim_{x \to 1^-} x^2 = 1^2 = 1,$$

$$\lim_{x \to 1^+} f(x) = \lim_{x \to 1^+} x^2 = 1^2 = 1.$$

Step 2: Check two-sided limit.
Since $\lim_{x \to 1^-} f(x) = \lim_{x \to 1^+} f(x) = 1$, the two-sided limit exists:

$$\lim_{x \to 1} f(x) = 1.$$

Step 3: Compare with function value.
We have $f(1) = 5$ (by definition), but $\lim_{x \to 1} f(x) = 1$. Since

$$\lim_{x \to 1} f(x) = 1 \neq 5 = f(1),$$

the function is **not continuous** at $x = 1$.

Interpretation and intuition: The function behaves like x^2 everywhere except at the single point $x = 1$, where it "jumps" to the value 5. Approaching from either side, we follow the parabola and expect to reach $(1, 1)$, but the function is artificially set to $(1, 5)$. This creates a "removable discontinuity"—we could make f continuous by redefining $f(1) = 1$.

Geometric visualization: Graph the parabola $y = x^2$ with an open circle at $(1, 1)$ and a filled circle at $(1, 5)$. The limit "wants" to go to $(1, 1)$ (where the parabola naturally leads), but the function value is isolated at $(1, 5)$.

Three conditions for continuity at a:

i. $f(a)$ is defined (yes, in our example)
ii. $\lim_{x \to a} f(x)$ exists (yes, in our example)
iii. $\lim_{x \to a} f(x) = f(a)$ (NO, fails in our example)

All three must hold for continuity.

Data science connection:

- **Model patching:** In deployment, a model might have a manually overridden output at specific inputs (e.g., blacklisting certain predictions). If this override doesn't match the model's natural output, it creates a discontinuity that can confuse downstream systems
- **Numerical implementations:** Functions like $\frac{\sin x}{x}$ need special handling at $x = 0$. If we define $f(0) = 0$ instead of $f(0) = 1$, we create this exact type of discontinuity (limit exists but doesn't match function value)
- **Caching and approximations:** Pre-computed lookup tables might store $f(a)$ incorrectly. If the cached value doesn't match $\lim_{x \to a} f(x)$, interpolation breaks down

- **Loss landscape:** Adding a penalty term only at specific parameter values (e.g., "+100 if $w_1 = 0$ else 0") creates discontinuities that gradient descent cannot handle

Common mistakes:

- Thinking "left limit = right limit" automatically implies continuity—must also check $f(a)$
- Confusing removable discontinuity (fixable by redefining one point) with jump discontinuity (left and right limits differ)
- Not checking all three conditions for continuity in order

Exercise 13: (Derivatives and Linearization — Intermediate) Analyze derivatives and first-order approximations.

a) Compute $f'(x)$ for $f(x) = e^{-0.2x^2} + 0.2\sin(2x)$ using the chain rule.

b) Construct the linear model $\ell_1(x) = f(x_0) + f'(x_0)(x - x_0)$ at $x_0 = 1$.

c) Numerically validate: compute $f(1 \pm 0.1)$ and $\ell_1(1 \pm 0.1)$, then find the absolute errors.

d) Error analysis: at what distance $|h|$ does the relative error $\frac{|f(1+h)-\ell_1(1+h)|}{|f(1)|}$ first exceed 5%?

Solution. Understanding the task: This exercise develops practical skills in computing derivatives, constructing linearizations, and understanding the locality of first-order approximations. These techniques are fundamental to gradient-based optimization, where we repeatedly build local linear models of objective functions to guide parameter updates.

a) **Computing the derivative $f'(x)$:**

Strategy: Apply the chain rule to each term separately, then combine using linearity of differentiation. For $e^{g(x)}$, the derivative is $e^{g(x)} \cdot g'(x)$. For $\sin(kx)$, it's $k\cos(kx)$.

Detailed computation:

Given $f(x) = e^{-0.2x^2} + 0.2\sin(2x)$.

Step 1: Differentiate the exponential term.

For $u(x) = e^{-0.2x^2}$, use the chain rule:

$$u'(x) = \frac{d}{dx}[e^{-0.2x^2}] = e^{-0.2x^2} \cdot \frac{d}{dx}[-0.2x^2] = e^{-0.2x^2} \cdot (-0.4x).$$

So $u'(x) = -0.4x\, e^{-0.2x^2}$.

Step 2: Differentiate the sine term.

For $v(x) = 0.2\sin(2x)$:

$$v'(x) = 0.2 \cdot \frac{d}{dx}[\sin(2x)] = 0.2 \cdot \cos(2x) \cdot 2 = 0.4\cos(2x).$$

Step 3: Combine using linearity.

Since $f(x) = u(x) + v(x)$:

$$f'(x) = u'(x) + v'(x) = -0.4x\, e^{-0.2x^2} + 0.4\cos(2x).$$

Final answer:

$$\boxed{f'(x) = -0.4x\, e^{-0.2x^2} + 0.4\cos(2x)}$$

Interpretation and intuition: The derivative combines two effects:

- The term $-0.4x\, e^{-0.2x^2}$ captures the rate of change from the Gaussian-like component. For $x > 0$, this is negative (decreasing), and the Gaussian envelope dampens the effect for large $|x|$
- The term $0.4\cos(2x)$ oscillates between ± 0.4, representing the periodic variation from the sine component

The interplay between these terms creates a function with local extrema influenced by both exponential decay and oscillation.

Data science connection:

- **Activation functions:** Combining smooth components (exponential, trigonometric) creates rich activation landscapes. Computing derivatives via chain rule is automatic in autodiff frameworks (PyTorch, JAX)
- **Loss landscapes:** Complex losses (e.g., with regularization, data augmentation) involve compositions. Understanding how chain rule propagates through layers is essential for backpropagation
- **Feature engineering:** Engineered features like $e^{-|x-\mu|^2/\sigma^2}$ (RBF kernels) or Fourier features require derivative computation for gradient-based feature selection

Common mistakes:

- Forgetting the chain rule: writing $\frac{d}{dx}[e^{-0.2x^2}] = e^{-0.2x^2}$ (missing the $-0.4x$ factor)
- Sign errors: the derivative of $-0.2x^2$ is $-0.4x$, not $+0.4x$
- Missing the constant factor: $\frac{d}{dx}[\sin(2x)] = 2\cos(2x)$, and then multiply by 0.2

b) **Constructing the linear model at $x_0 = 1$:**

Strategy: The first-order Taylor approximation (tangent line) at x_0 is $\ell_1(x) = f(x_0) + f'(x_0)(x - x_0)$. We need to evaluate both f and f' at $x_0 = 1$.

Detailed computation:

Step 1: Evaluate $f(1)$.

$$f(1) = e^{-0.2(1)^2} + 0.2\sin(2 \cdot 1) = e^{-0.2} + 0.2\sin(2).$$

Numerically:

$$\begin{aligned} e^{-0.2} &\approx 0.8187, \\ \sin(2) &\approx 0.9093, \\ f(1) &\approx 0.8187 + 0.2(0.9093) \approx 0.8187 + 0.1819 \approx 1.0006. \end{aligned}$$

Step 2: Evaluate $f'(1)$.

$$f'(1) = -0.4(1)\, e^{-0.2(1)^2} + 0.4\cos(2 \cdot 1) = -0.4\, e^{-0.2} + 0.4\cos(2).$$

Numerically:

$$\begin{aligned} e^{-0.2} &\approx 0.8187, \\ \cos(2) &\approx -0.4161, \\ f'(1) &\approx -0.4(0.8187) + 0.4(-0.4161) \approx -0.3275 - 0.1664 \approx -0.4939. \end{aligned}$$

Step 3: Construct $\ell_1(x)$.

$$\ell_1(x) = f(1) + f'(1)(x - 1) \approx 1.0006 - 0.4939(x - 1).$$

Final answer:

$$\boxed{\ell_1(x) \approx 1.0006 - 0.4939(x - 1)}$$

Or equivalently, expanding:

$$\ell_1(x) \approx -0.4939x + 1.4945.$$

Interpretation and intuition: The linear model is a tangent line to f at $x = 1$. The slope $f'(1) \approx -0.494$ is negative, indicating f is decreasing at $x = 1$. The intercept $f(1) \approx 1.001$ is the starting point of the tangent.

Geometric visualization: If you zoom in on the graph of f near $x = 1$, it looks like a straight line with slope ≈ -0.494. This is the essence of differentiability: "local straightness."

Data science connection:

- **Gradient descent step:** The update $x_{k+1} = x_k - \eta f'(x_k)$ uses the local linear approximation. At $x_k = 1$ with $\eta = 0.1$, we'd move to $x_{k+1} = 1 - 0.1(-0.494) \approx 1.049$, stepping "uphill" along the negative gradient (since we're minimizing $-f$ or maximizing f)
- **Newton's method:** First-order methods use ℓ_1 to approximate; second-order methods add curvature. Understanding the linear model is the first step
- **Sensitivity analysis:** $f'(1) \approx -0.494$ means a small increase in x from 1 causes f to decrease by approximately half that amount. This quantifies local sensitivity

Common mistakes:

- Using $\ell_1(x) = f'(1) \cdot x$ (forgetting the $f(1)$ offset and $(x - x_0)$ shift)
- Sign errors in evaluation: $\cos(2)$ is negative, not positive
- Not recognizing this as the equation of a line: $y = mx + b$ form

c) **Numerical validation:**

Strategy: Evaluate both the true function f and the linear approximation ℓ_1 at the test points $x = 1 \pm 0.1$, then compute absolute errors $|f(x) - \ell_1(x)|$.

Detailed computation:

Test point 1: $x = 0.9$ (i.e., $h = -0.1$):

True function value:

$$\begin{aligned} f(0.9) &= e^{-0.2(0.9)^2} + 0.2\sin(2 \cdot 0.9) \\ &= e^{-0.162} + 0.2\sin(1.8) \\ &\approx 0.8504 + 0.2(0.9738) \approx 0.8504 + 0.1948 \approx 1.0452. \end{aligned}$$

Linear approximation:

$$\ell_1(0.9) = 1.0006 - 0.4939(0.9 - 1) = 1.0006 - 0.4939(-0.1) = 1.0006 + 0.0494 \approx 1.0500.$$

Absolute error:

$$|f(0.9) - \ell_1(0.9)| \approx |1.0452 - 1.0500| \approx 0.0048.$$

Test point 2: $x = 1.1$ *(i.e.,* $h = 0.1$*):*
True function value:

$$\begin{aligned} f(1.1) &= e^{-0.2(1.1)^2} + 0.2\sin(2 \cdot 1.1) \\ &= e^{-0.242} + 0.2\sin(2.2) \\ &\approx 0.7852 + 0.2(0.8085) \approx 0.7852 + 0.1617 \approx 0.9469. \end{aligned}$$

Linear approximation:

$$\ell_1(1.1) = 1.0006 - 0.4939(1.1 - 1) = 1.0006 - 0.4939(0.1) = 1.0006 - 0.0494 \approx 0.9512.$$

Absolute error:

$$|f(1.1) - \ell_1(1.1)| \approx |0.9469 - 0.9512| \approx 0.0043.$$

Summary table:

x	$f(x)$	$\ell_1(x)$	$\lvert f(x) - \ell_1(x) \rvert$
0.9	1.0452	1.0500	0.0048
1.1	0.9469	0.9512	0.0043

Interpretation and intuition: The errors are quite small (less than 0.5% of the function value), confirming that the linear approximation is accurate near $x_0 = 1$ for $|h| = 0.1$. Both test points show similar error magnitudes, suggesting the approximation is symmetric (though curvature effects would typically break this symmetry for larger $|h|$).

Visual understanding: If we plot f and ℓ_1 on the interval $[0.9, 1.1]$, the tangent line would nearly coincide with the curve. The vertical gap between them (the error) is barely visible at this scale.

Data science connection:

- **Gradient descent validation:** After taking a step $x_{k+1} = x_k - \eta \nabla f(x_k)$, we can check if $f(x_{k+1})$ matches the predicted decrease from the linear model. Large discrepancies suggest the step size is too large or the loss is highly curved

- **Trust region methods:** These methods explicitly construct a region (e.g., $|x - x_0| \leq \Delta$) where the linear (or quadratic) model is trusted. Our numerical validation shows that for $|h| \leq 0.1$, the linear model has ¡0.5% error
- **Learning rate selection:** If the linear model predicts $f(x - \eta f'(x)) \approx f(x) - \eta(f'(x))^2$ but the actual decrease is much less, we should reduce η

Common mistakes:

- Computing relative error as $\frac{|f(x)-\ell_1(x)|}{|\ell_1(x)|}$ instead of comparing to true function value or using absolute error
- Not recognizing that small absolute errors can be large relative errors if $|f(x)|$ is small
- Expecting perfect agreement—the linearization is an approximation

d) **Error analysis and validity radius:**

Strategy: We seek the smallest $|h|$ where the relative error exceeds 5%. This requires computing $f(1+h)$ and $\ell_1(1+h)$ for various h values and checking when

$$\frac{|f(1+h) - \ell_1(1+h)|}{|f(1)|} > 0.05.$$

Detailed analysis:

Step 1: Set up the error function.

Define the relative error as

$$E(h) = \frac{|f(1+h) - \ell_1(1+h)|}{|f(1)|}.$$

We know:

- $f(1) \approx 1.0006$
- $\ell_1(1+h) = f(1) + f'(1) \cdot h \approx 1.0006 - 0.4939h$
- $f(1+h) = e^{-0.2(1+h)^2} + 0.2\sin(2(1+h))$

Step 2: Compute for increasing $|h|$.

Let's test several values:

For $h = 0.2$:

$$\begin{aligned} f(1.2) &\approx e^{-0.288} + 0.2\sin(2.4) \approx 0.7498 + 0.2(0.6755) \approx 0.8849, \\ \ell_1(1.2) &\approx 1.0006 - 0.4939(0.2) \approx 1.0006 - 0.0988 \approx 0.9018, \\ E(0.2) &\approx \frac{|0.8849 - 0.9018|}{1.0006} \approx \frac{0.0169}{1.0006} \approx 0.0169 \approx 1.69\%. \end{aligned}$$

For $h = 0.3$:

$$\begin{aligned} f(1.3) &\approx e^{-0.338} + 0.2\sin(2.6) \approx 0.7130 + 0.2(0.5155) \approx 0.8161, \\ \ell_1(1.3) &\approx 1.0006 - 0.4939(0.3) \approx 1.0006 - 0.1482 \approx 0.8524, \\ E(0.3) &\approx \frac{|0.8161 - 0.8524|}{1.0006} \approx \frac{0.0363}{1.0006} \approx 0.0363 \approx 3.63\%. \end{aligned}$$

For $h = 0.4$:

$$f(1.4) \approx e^{-0.392} + 0.2\sin(2.8) \approx 0.6757 + 0.2(0.3350) \approx 0.7427,$$
$$\ell_1(1.4) \approx 1.0006 - 0.4939(0.4) \approx 1.0006 - 0.1976 \approx 0.8030,$$
$$E(0.4) \approx \frac{|0.7427 - 0.8030|}{1.0006} \approx \frac{0.0603}{1.0006} \approx 0.0603 \approx 6.03\%.$$

Step 3: Interpolate.
We see that:

- At $|h| = 0.3$: error $\approx 3.63\%$ (below 5%)
- At $|h| = 0.4$: error $\approx 6.03\%$ (exceeds 5%)

The 5% threshold is crossed somewhere between $h = 0.3$ and $h = 0.4$. Linear interpolation gives approximately:

$$h^* \approx 0.3 + (0.4 - 0.3) \cdot \frac{5\% - 3.63\%}{6.03\% - 3.63\%} \approx 0.3 + 0.1 \cdot \frac{1.37}{2.40} \approx 0.3 + 0.057 \approx 0.36.$$

Conclusion: The relative error first exceeds 5% at approximately $\boxed{|h| \approx 0.36}$ (or roughly when $|x - 1| \gtrsim 0.36$).
Interpretation and intuition: The linearization is accurate (within 5% relative error) for $x \in [0.64, 1.36]$, an interval of width ≈ 0.72 centered at $x_0 = 1$. Beyond this range, curvature effects (from the exponential decay and sine oscillation) dominate, and the tangent line increasingly deviates from the true function.
Visual understanding: Imagine plotting both f and ℓ_1. They're nearly indistinguishable for $x \in [0.9, 1.1]$, but as we move further from $x = 1$, the curves separate. At $x \approx 0.64$ or $x \approx 1.36$, the vertical gap reaches 5% of $f(1)$.
Data science connection:

- **Step-size selection:** If we're at $x = 1$ and want the linear model to be accurate within 5%, we should take steps with $|\eta f'(1)| \lesssim 0.36$. Since $|f'(1)| \approx 0.494$, this suggests $\eta \lesssim 0.36/0.494 \approx 0.73$
- **Trust region radius:** In trust region methods, the radius Δ should be chosen so the model error stays below a tolerance. Here, $\Delta \approx 0.36$ gives 5% accuracy
- **Curvature and second-order methods:** The breakdown of the linear model signals that curvature (second derivatives) matters. For $|h| > 0.36$, a quadratic model $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$ would be more accurate
- **Learning rate schedules:** As optimization progresses and we approach a minimum, the effective "curvature" increases. The valid radius of the linear approximation shrinks, necessitating smaller learning rates

Common mistakes:

- Using absolute error instead of relative error—5% relative is problem-dependent
- Not interpolating to find the exact threshold—testing only integer multiples of 0.1 is insufficient

- Forgetting that the validity radius depends on the function's curvature—smooth functions have larger radii

Exercise 14: (ML-Flavored Applications — Intermediate) Apply first-order calculus to optimization.

a) For a differentiable function $f : \mathbb{R} \to \mathbb{R}$ and small $\eta > 0$, use the first-order Taylor model to show

$$f(x - \eta f'(x)) \approx f(x) - \eta(f'(x))^2.$$

Explain why this guarantees monotone decrease when $f'(x) \neq 0$ and η is sufficiently small.

b) Suppose $|f'(x)| \leq L$ for all x in some interval. Derive a step-size bound $\eta < \eta_{\max}$ that ensures the approximation error $\left|f(x - \eta f'(x)) - [f(x) - \eta(f'(x))^2]\right|$ is small relative to the predicted decrease.

c) Given the quadratic loss $f(w) = \|Xw - y\|_2^2$ where $X \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$, compute $\nabla f(w)$ and explain how $\left|\frac{\partial f}{\partial w_i}(w)\right|$ measures the sensitivity of the loss to changes in the i-th feature weight.

Solution. Understanding the task: This exercise connects first-order calculus to gradient descent, step-size selection, and feature sensitivity analysis. These are core concepts in machine learning optimization, providing the mathematical foundation for training neural networks and other parametric models.

a) **Monotone decrease via first-order model:**

Strategy: We'll apply the first-order Taylor approximation $f(x + h) \approx f(x) + f'(x)h$ with $h = -\eta f'(x)$, then analyze when the approximation predicts a decrease.

Detailed proof:

Step 1: Apply Taylor's theorem.

For differentiable f and small h, the first-order Taylor expansion gives:

$$f(x + h) = f(x) + f'(x)h + o(|h|),$$

where $o(|h|)$ denotes a remainder that vanishes faster than $|h|$ as $h \to 0$.

Step 2: Substitute $h = -\eta f'(x)$.

Let $h = -\eta f'(x)$ where $\eta > 0$ is a small step size. Then:

$$f(x - \eta f'(x)) = f(x) + f'(x) \cdot (-\eta f'(x)) + o(\left|-\eta f'(x)\right|).$$

Simplifying:

$$f(x - \eta f'(x)) = f(x) - \eta(f'(x))^2 + o(\eta \left|f'(x)\right|).$$

Step 3: Neglect higher-order terms.

For sufficiently small η, the remainder $o(\eta\,|f'(x)|)$ is negligible compared to the linear term $\eta(f'(x))^2$. Thus:

$$\boxed{f(x-\eta f'(x)) \approx f(x) - \eta(f'(x))^2}$$

Step 4: Analyze the decrease.
The change in function value is:

$$f(x-\eta f'(x)) - f(x) \approx -\eta(f'(x))^2.$$

Since $(f'(x))^2 \geq 0$ and $\eta > 0$, we have:

$$f(x-\eta f'(x)) - f(x) \leq 0,$$

with strict inequality when $f'(x) \neq 0$.
Conclusion: For small enough $\eta > 0$ and $f'(x) \neq 0$, the update $x \mapsto x - \eta f'(x)$ strictly decreases f.
Interpretation and intuition:

- The term $-\eta(f'(x))^2$ is the predicted decrease. Its magnitude depends on:
  - Step size η: larger steps $\rightarrow$ larger predicted decrease
  - Gradient squared $(f'(x))^2$: steeper slopes $\rightarrow$ larger decrease
- The direction $-f'(x)$ is the direction of steepest descent (opposite the gradient)
- "Sufficiently small η" ensures we stay in the region where the linear model is accurate

Geometric visualization: Imagine standing on a hillside (the graph of f). The tangent line at your position has slope $f'(x)$. Moving a small distance η in the downhill direction $-f'(x)$ takes you to a lower elevation, with the drop proportional to $\eta(f'(x))^2$.
Data science connection:

- **Gradient descent:** This result is the foundation of gradient descent: $w_{t+1} = w_t - \eta\nabla f(w_t)$. The loss decreases by approximately $\eta\,\|\nabla f(w_t)\|_2^2$ per iteration
- **Step-size selection:** If η is too large, the linear approximation breaks down and we might overshoot. If too small, convergence is slow. The optimal η balances speed and accuracy
- **Convergence diagnostics:** Monitor the actual decrease $f(w_{t+1}) - f(w_t)$ vs. the predicted decrease $-\eta\,\|\nabla f(w_t)\|_2^2$. Large discrepancies suggest η is too large or the loss has high curvature
- **Adaptive methods:** Algorithms like Adam, RMSProp adaptively scale η based on gradient history to maintain stable progress

Common mistakes:

- Forgetting the square: the decrease is $\eta(f'(x))^2$, not $\eta\,|f'(x)|$
- Thinking decrease is guaranteed for any η—must be "sufficiently small" for the linear approximation to hold

- Not recognizing that $f'(x) = 0$ (critical point) gives zero decrease—gradient descent stalls

b) **Step-size bound from Lipschitz constant:**

Strategy: If $|f'(x)| \leq L$ (Lipschitz continuous derivative), the second-order remainder can be bounded, allowing us to derive a step-size condition that ensures the error is small relative to the predicted decrease.

Detailed analysis:

Step 1: Taylor's theorem with remainder.

By the mean value theorem, for some ξ between x and $x - \eta f'(x)$:

$$f(x - \eta f'(x)) = f(x) - \eta(f'(x))^2 + \frac{1}{2}f''(\xi)[\eta f'(x)]^2.$$

The error term is:

$$E = \frac{1}{2}f''(\xi)\eta^2(f'(x))^2.$$

Step 2: Bound f'' using Lipschitz condition.

If $|f'(x)| \leq L$ for all x, then by the fundamental theorem of calculus:

$$\left|f''(x)\right| \leq L \quad \text{(crude bound)}.$$

More precisely, if f' is L-Lipschitz (i.e., $|f'(x) - f'(y)| \leq L\,|x - y|$), then $|f''(x)| \leq L$ almost everywhere.

Thus:

$$|E| \leq \frac{1}{2}L\eta^2(f'(x))^2.$$

Step 3: Require error to be small relative to predicted decrease.

The predicted decrease is $D = \eta(f'(x))^2$. We want:

$$\frac{|E|}{D} \leq c \quad \text{for some tolerance } c < 1.$$

Substituting:

$$\frac{\frac{1}{2}L\eta^2(f'(x))^2}{\eta(f'(x))^2} = \frac{L\eta}{2} \leq c.$$

Solving for η:

$$\eta \leq \frac{2c}{L}.$$

Step 4: Common choice.

For robustness, a typical choice is $c = \frac{1}{2}$ (error is at most half the predicted decrease), giving:

$$\boxed{\eta < \frac{1}{L}}$$

as a conservative step-size bound.

Interpretation and intuition:

- The Lipschitz constant L measures how fast the gradient can change. Large L means high curvature, requiring smaller steps

- The bound $\eta < 1/L$ ensures we don't "overstep"—the linear model remains accurate
- For smooth, strongly convex functions, this gives linear convergence: $f(w_t) - f(w^*) \leq (1 - \eta\mu)^t[f(w_0) - f(w^*)]$ where μ is the strong convexity constant

Geometric visualization: If f has high curvature (large L), the tangent line quickly deviates from the function. Small steps are needed to stay near the tangent. If f is nearly flat (small L), larger steps are safe.

Data science connection:

- **Smooth losses:** Cross-entropy and MSE have bounded gradients under typical conditions. Estimating L (e.g., from largest eigenvalue of Hessian) guides η selection
- **Lipschitz regularization:** Spectral normalization in GANs constrains L for discriminator, stabilizing training
- **Line search:** Backtracking line search adaptively finds η satisfying the Armijo condition: $f(w - \eta\nabla f(w)) \leq f(w) - c\eta \|\nabla f(w)\|^2$
- **Learning rate warmup:** Early in training, L may be large (far from optimum). Gradually increasing η from 0 to $\frac{1}{L}$ prevents early instability

Common mistakes:

- Using $\eta > 1/L$—this can cause oscillation or divergence
- Not recognizing that L can vary across the domain—may need adaptive η
- Confusing Lipschitz constant of f vs. f'—we need the latter for gradient descent analysis

c) **Gradient and feature sensitivity for quadratic loss:**

Strategy: Expand the squared norm, compute the gradient component-wise, and interpret the magnitude of each partial derivative as measuring sensitivity to the corresponding feature weight.

Detailed computation:

Given $f(w) = \|Xw - y\|_2^2$ where $X \in \mathbb{R}^{n \times d}$, $w \in \mathbb{R}^d$, $y \in \mathbb{R}^n$.

Step 1: Expand the norm.

$$f(w) = (Xw - y)^\top (Xw - y) = w^\top X^\top Xw - 2w^\top X^\top y + y^\top y.$$

Step 2: Compute the gradient.

Using matrix calculus:

$$\nabla f(w) = \nabla[w^\top X^\top Xw - 2w^\top X^\top y + y^\top y].$$

Apply the rules:

- $\nabla_w[w^\top Aw] = 2Aw$ for symmetric A
- $\nabla_w[w^\top b] = b$
- $\nabla_w[c] = 0$ for constant c

Since $X^\top X$ is symmetric:

$$\nabla f(w) = 2X^\top Xw - 2X^\top y = 2X^\top (Xw - y).$$

Final answer:

$$\boxed{\nabla f(w) = 2X^\top (Xw - y)}$$

Step 3: Interpret the i-th component.
The i-th partial derivative is:

$$\frac{\partial f}{\partial w_i}(w) = [\nabla f(w)]_i = 2[X^\top (Xw - y)]_i = 2\sum_{j=1}^{n} X_{ji}(Xw - y)_j.$$

Equivalently, if x_i denotes the i-th column of X:

$$\frac{\partial f}{\partial w_i}(w) = 2\langle x_i, Xw - y\rangle = 2x_i^\top (Xw - y).$$

Interpretation and intuition:

- The gradient component $\frac{\partial f}{\partial w_i}$ measures how much the loss changes when we adjust w_i
- It's the inner product of the i-th feature vector x_i with the residual $Xw - y$
- Large $\left|\frac{\partial f}{\partial w_i}\right|$ means:
  - The i-th feature is highly correlated with the current residual
  - Adjusting w_i would have a large impact on the loss
  - The i-th feature is "important" for explaining the residual
- Small $\left|\frac{\partial f}{\partial w_i}\right|$ suggests the i-th feature is:
  - Already well-fitted (residual is orthogonal to x_i)
  - Or not useful for explaining the remaining error

Geometric visualization: The gradient $\nabla f(w)$ points in the direction of steepest increase. Each component $\frac{\partial f}{\partial w_i}$ is the slope along the i-th coordinate axis. Moving w_i by Δw_i changes the loss by approximately $2x_i^\top (Xw - y) \cdot \Delta w_i$.

Data science connection:

- **Feature importance:** In linear regression, $\left|\frac{\partial f}{\partial w_i}\right|$ at the current iterate quantifies the i-th feature's contribution to the gradient. Features with consistently large gradients are more influential
- **Feature selection:** Regularization methods (LASSO, elastic net) use gradient information to decide which weights to shrink. Features with small $\left|\frac{\partial f}{\partial w_i}\right|$ are candidates for elimination
- **Saliency maps:** In neural networks, $\left|\frac{\partial \mathcal{L}}{\partial x_i}\right|$ (gradient w.r.t. input) measures pixel/feature importance for predictions. High magnitude indicates sensitivity
- **Gradient-based optimization:** The update $w_i^{(t+1)} = w_i^{(t)} - \eta \frac{\partial f}{\partial w_i}(w^{(t)})$ adjusts each weight proportionally to its gradient, naturally prioritizing influential features
- **Diagnosis:** If $\|\nabla f(w)\|$ is small but loss is high, we're near a local minimum or saddle point. If certain $\left|\frac{\partial f}{\partial w_i}\right|$ remain large, those features need more attention

Common mistakes:

- Forgetting the factor of 2: $\nabla[w^\top Aw] = 2Aw$ for symmetric A
- Confusing gradient w.r.t. weights ($\nabla_w f$) with gradient w.r.t. inputs ($\nabla_x f$)—different interpretations
- Thinking large $\left|\frac{\partial f}{\partial w_i}\right|$ always means important feature—could also indicate w_i is poorly tuned
- Not recognizing that gradient is $2X^\top(\text{residual})$—this form is crucial for interpreting the normal equations $X^\top Xw = X^\top y$ (where gradient vanishes)