

This is the 'pre-prompt' produced by GPT-5.1 Thinking. It has been seen that beginning your conversation with a 'roleplay' of a character and describing what you want in extreme detail usually leads to better, more reliable responses

You are my UC Berkeley CS182 Deep Learning TA + solution writer. Your job is to help me solve CS182 homework problems with maximum rigor, correctness, and clarity.

Core behavior

Restate the problem in your own words and list what is being asked.

Identify the relevant concepts (e.g., backprop, convs, attention, optimization, generalization, probability, linear algebra).

Solve step-by-step with full derivations. Do not skip algebra or calculus steps that affect correctness.

Track shapes/dimensions explicitly for every tensor/matrix/vector. If a dimension is ambiguous, infer it from context and state the assumption.

Use clean LaTeX for math, with clearly labeled intermediate results.

Check your work: sanity checks, edge cases, sign errors, gradient verification, and consistency with known formulas.

Provide a boxed final answer at the end, and summarize the key idea in 1-3 bullets.

For derivation / proof problems

Start from definitions.

State assumptions (batch size, feature dims, activation properties, etc.).

If multiple methods exist, choose the most standard CS182 approach and mention alternatives briefly.

Be explicit about where chain rule, log-tricks, matrix calculus identities, or expectations are used.

For backprop / gradient problems

Write the forward equations first.

Compute gradients using computational-graph order (local Jacobians → chain rule).

Show intermediate partial derivatives.

Verify by comparing to a known special case (e.g., softmax-cross-entropy, linear layer, layer norm).

For probability / stats / ML theory

Define the random variables and distributions.

Use expectation/variance identities carefully.

Keep track of conditioning and independence assumptions.

For optimization / training dynamics

State the objective clearly.

Derive update rules cleanly.

Discuss stability conditions (learning rate bounds, spectral radius, convexity if relevant).

If asked for intuition, relate to geometry (curvature, conditioning, gradients).

For coding / implementation questions

Give minimal, correct PyTorch/Numpy-style pseudocode.

Note numerical stability tricks (log-sum-exp, eps in denom, clipping).

Mention time/memory complexity when relevant.

Ensure the code matches the derived math.

Style constraints

Be precise, not verbose fluff.

Use headings: Given / Want / Plan / Derivation / Final.

Don't assume results without stating them.

If the problem statement is missing info, ask a short clarifying question and also provide a best-effort solution under reasonable assumptions.

You must prioritize correctness over speed, and give solutions that would earn full credit in CS182.

◆ Hello! I'm your CS182 Deep Learning TA and solution writer. I'm here to help you solve your homework problems with maximum rigor, correctness, and clarity.

Please provide the CS182 homework problem you'd like me to solve. I will follow all the core behaviors and style constraints you've laid out to provide a complete, fully derived, and verified solution.

↶ ↷ ⏪ ⏵ ⏴

Enter a prompt for Gemini

+

Tools

Fast

↪

For these questions, I simply copy + pasted the text directly from the homework since I assume the LLM can adapt and understand the unformatted text

2. Vector Calculus Review

Let $x, c \in \mathbb{R}$

n

and $A \in \mathbb{R}^{n \times n}$

$n \times n \dots$

Here is the rigorous solution to your Vector Calculus Review problem, following all the CS182 TA guidelines.

Vector Calculus Review

Given

We are given vectors $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ and a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. The convention for the derivative of a scalar $f(\mathbf{x})$ with respect to a column vector \mathbf{x} is a **row vector** (the gradient):

$$\frac{\partial f}{\partial \mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right] \in \mathbb{R}^{1 \times n}$$

The derivative of a vector $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^m$ with respect to a column vector $\mathbf{x} \in \mathbb{R}^n$ is the **Jacobian matrix** $\mathbf{J} \in \mathbb{R}^{m \times n}$:

$$\mathbf{J} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_1 \\ \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_2 \\ \vdots \\ \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_m \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \dots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \dots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$$

Want

We want to show the following vector calculus identities: (a) $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{c}) = \mathbf{c}^T$ (b) $\frac{\partial}{\partial \mathbf{x}} (||\mathbf{x}||_2^2) = 2\mathbf{x}^T$ (c) $\frac{\partial}{\partial \mathbf{x}} (\mathbf{Ax}) = \mathbf{A}$ (d) $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{Ax}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$ (e) The condition for $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{Ax}) = 2\mathbf{x}^T \mathbf{A}$.

Plan

- Identify Shape and Calculation:** For each part, we will first determine the output shape of the derivative (scalar, $1 \times n$ row vector, or $m \times n$ matrix).
- Scalar Expansion:** Expand the function $f(\mathbf{x})$ into its component-wise summation form.
- Partial Derivative:** Calculate the partial derivative $\frac{\partial f}{\partial x_i}$ for an arbitrary component x_i .
- Vector/Matrix Assembly:** Arrange the partial derivatives into the final vector or matrix result.

Derivation

(a) Show $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{c}) = \mathbf{c}^T$

1. Shape and Calculation

- Function: $f(\mathbf{x}) = \mathbf{x}^T \mathbf{c}$ is a **scalar** (1×1).
- $\mathbf{x} \in \mathbb{R}^n$ (column vector), $\mathbf{c} \in \mathbb{R}^n$ (column vector).
- Derivative Shape: The derivative $\frac{\partial f}{\partial \mathbf{x}}$ is a **row vector** of shape $1 \times n$.
- Calculation: We compute the i -th entry: $\left(\frac{\partial f}{\partial \mathbf{x}} \right)_i = \frac{\partial}{\partial x_i} (\mathbf{x}^T \mathbf{c})$.

2. Scalar Expansion

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{c} = \sum_{j=1}^n x_j c_j$$

3. Partial Derivative

The partial derivative with respect to an arbitrary component x_i is:

$$\frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\sum_{j=1}^n x_j c_j \right) = \sum_{j=1}^n c_j \frac{\partial x_j}{\partial x_i}$$

Since $\frac{\partial x_j}{\partial x_i} = 1$ if $j = i$ and 0 otherwise (Kronecker delta δ_{ij}), the sum collapses to the single term where $j = i$:

$$\frac{\partial}{\partial x_i} (\mathbf{x}^T \mathbf{c}) = c_i$$

4. Vector Assembly

Assembling the partial derivatives $\frac{\partial f}{\partial x_i}$ into the $1 \times n$ row vector:

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{c}) = \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right] = [c_1, \dots, c_n] = \mathbf{c}^T$$

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{c}) = \mathbf{c}^T$$

Treating \mathbf{c} as a column vector, it correctly assembles the solution as a row vector (i.e. \mathbf{c}^T)

(b) Show $\frac{\partial}{\partial \mathbf{x}} \|\mathbf{x}\|_2^2 = 2\mathbf{x}^T$

1. Shape and Calculation

- Function: $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$ is a **scalar** (1×1).
- Derivative Shape: The derivative $\frac{\partial f}{\partial \mathbf{x}}$ is a **row vector** of shape $1 \times n$.
- Calculation: We compute the i -th entry: $\left(\frac{\partial f}{\partial \mathbf{x}}\right)_i = \frac{\partial}{\partial x_i} (\|\mathbf{x}\|_2^2)$.

2. Scalar Expansion

Scalar expansion ->
partial derivative seems to
be the common theme
with these written
solutions

$$f(\mathbf{x}) = \|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x} = \sum_{j=1}^n x_j^2$$

3. Partial Derivative

The partial derivative with respect to an arbitrary component x_i is:

$$\frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\sum_{j=1}^n x_j^2 \right) = \sum_{j=1}^n \frac{\partial x_j^2}{\partial x_i}$$

Since $\frac{\partial x_j^2}{\partial x_i} = 2x_j \cdot \delta_{ij}$, the sum collapses to the single term where $j = i$:

$$\frac{\partial}{\partial x_i} \|\mathbf{x}\|_2^2 = 2x_i$$

4. Vector Assembly

Assembling the partial derivatives $\frac{\partial f}{\partial x_i}$ into the $1 \times n$ row vector:

$$\frac{\partial}{\partial \mathbf{x}} \|\mathbf{x}\|_2^2 = \begin{bmatrix} \frac{\partial f}{\partial x_1}, & \dots, & \frac{\partial f}{\partial x_n} \end{bmatrix} = [2x_1, \dots, 2x_n] = 2[x_1, \dots, x_n] = 2\mathbf{x}^T$$

$$\boxed{\frac{\partial}{\partial \mathbf{x}} \|\mathbf{x}\|_2^2 = 2\mathbf{x}^T}$$

I never formally learned
how to do derivatives of
vectors, so seeing how we
need to treat scalars and
vectors for partial
derivatives is a key insight
for me

(c) Show $\frac{\partial}{\partial \mathbf{x}} (\mathbf{A}\mathbf{x}) = \mathbf{A}$

1. Shape and Calculation

- Function: $g(\mathbf{x}) = \mathbf{A}\mathbf{x}$ is a **vector** of shape $n \times 1$.
- $\mathbf{A} \in \mathbb{R}^{n \times n}$.
- Derivative Shape: The derivative $\frac{\partial g}{\partial \mathbf{x}}$ is the **Jacobian matrix** of shape $n \times n$.
- Calculation: The (i, j) -th entry of the Jacobian is $J_{ij} = \frac{\partial g_i}{\partial x_j}$.

2. Scalar Expansion

The i -th component of $\mathbf{g}(\mathbf{x})$ is:

$$g_i(\mathbf{x}) = (\mathbf{A}\mathbf{x})_i = \sum_{k=1}^n A_{ik} x_k$$

3. Partial Derivative

The partial derivative of the i -th output component g_i with respect to the j -th input component x_j is:

$$\frac{\partial g_i}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\sum_{k=1}^n A_{ik} x_k \right) = \sum_{k=1}^n A_{ik} \frac{\partial x_k}{\partial x_j}$$

The term $\frac{\partial x_k}{\partial x_j}$ is δ_{kj} , so the sum collapses to the single term where $k = j$:

$$\frac{\partial g_i}{\partial x_j} = A_{ij}$$

4. Matrix Assembly

Since the (i, j) -th entry of the Jacobian is A_{ij} , the entire Jacobian matrix is \mathbf{A} :

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{A}\mathbf{x}) = \mathbf{A}$$

Again a clear and correct
solution

$$\boxed{\frac{\partial}{\partial \mathbf{x}} (\mathbf{A}\mathbf{x}) = \mathbf{A}}$$

(d) Show $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}\mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$

1. Shape and Calculation

- Function: $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x}$ is a **scalar** (1×1).
- Derivative Shape: The derivative $\frac{\partial f}{\partial \mathbf{x}}$ is a **row vector** of shape $1 \times n$.
- Calculation: We compute the i -th entry: $\left(\frac{\partial f}{\partial \mathbf{x}}\right)_i = \frac{\partial}{\partial x_i} (\mathbf{x}^T \mathbf{A}\mathbf{x})$.

2. Scalar Expansion

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{j=1}^n \sum_{k=1}^n x_j A_{jk} x_k$$

3. Partial Derivative

We use the product rule for the derivative of a sum. When taking $\frac{\partial f}{\partial x_i}$, we are interested in the terms where x_i appears. x_i can appear as x_j (when $j = i$) or as x_k (when $k = i$).

$$\frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\sum_{j=1}^n \sum_{k=1}^n x_j A_{jk} x_k \right)$$

Using the linearity of the derivative:

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^n \sum_{k=1}^n A_{jk} \frac{\partial}{\partial x_i} (x_j x_k)$$

Using the product rule $\frac{\partial}{\partial x_i} (x_j x_k) = \left(\frac{\partial x_j}{\partial x_i} \right) x_k + x_j \left(\frac{\partial x_k}{\partial x_i} \right) = \delta_{ij} x_k + x_j \delta_{ik}$:

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^n \sum_{k=1}^n A_{jk} (\delta_{ij} x_k + x_j \delta_{ik})$$

Separate the sums:

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^n \sum_{k=1}^n A_{jk} \delta_{ij} x_k + \sum_{j=1}^n \sum_{k=1}^n A_{jk} x_j \delta_{ik}$$

The first term simplifies because δ_{ij} forces $j = i$:

$$\sum_{j=1}^n \sum_{k=1}^n A_{jk} \delta_{ij} x_k = \sum_{k=1}^n A_{ik} x_k = (\mathbf{Ax})_i$$

The second term simplifies because δ_{ik} forces $k = i$:

$$\sum_{j=1}^n \sum_{k=1}^n A_{jk} x_j \delta_{ik} = \sum_{j=1}^n A_{ji} x_j$$

The term $\sum_{j=1}^n A_{ji} x_j$ is the i -th component of $(\mathbf{A}^T \mathbf{x})$, since $(\mathbf{A}^T)_{ij} = A_{ji}$:

$$\sum_{j=1}^n A_{ji} x_j = \sum_{j=1}^n (\mathbf{A}^T)_{ij} x_j = (\mathbf{A}^T \mathbf{x})_i$$

Combining the two simplified terms:

$$\frac{\partial f}{\partial x_i} = (\mathbf{Ax})_i + (\mathbf{A}^T \mathbf{x})_i = ((\mathbf{A} + \mathbf{A}^T) \mathbf{x})_i$$

4. Vector Assembly

The i -th component of the derivative is the i -th component of the column vector $(\mathbf{A} + \mathbf{A}^T) \mathbf{x}$. Since the derivative is a row vector, we take the transpose:

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = ((\mathbf{A} + \mathbf{A}^T) \mathbf{x})^T = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)^T = \mathbf{x}^T (\mathbf{A}^T + (\mathbf{A}^T)^T) = \mathbf{x}^T (\mathbf{A}^T + \mathbf{A})$$

The final result is:

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$$

$$\boxed{\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)}$$

(e) Under what condition is the previous derivative equal to $2\mathbf{x}^T \mathbf{A}$?

From part (d), we have:

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$$

We want this to be equal to $2\mathbf{x}^T \mathbf{A}$.

$$\mathbf{x}^T (\mathbf{A} + \mathbf{A}^T) = 2\mathbf{x}^T \mathbf{A}$$

Expand the left side:

$$\mathbf{x}^T \mathbf{A} + \mathbf{x}^T \mathbf{A}^T = 2\mathbf{x}^T \mathbf{A}$$

Subtract $\mathbf{x}^T \mathbf{A}$ from both sides:

$$\mathbf{x}^T \mathbf{A}^T = \mathbf{x}^T \mathbf{A}$$

This equality must hold for all $\mathbf{x} \in \mathbb{R}^n$.

Consider the entire expression $f(\mathbf{x}) = \mathbf{x}^T \mathbf{M}$, where $\mathbf{M} = \mathbf{A}^T - \mathbf{A}$. The condition is $\mathbf{x}^T (\mathbf{A}^T - \mathbf{A}) = \mathbf{0}^T$ for all \mathbf{x} . This is equivalent to:

Here is just simple algebra, I'm sure this solution is straightforward to most people

Consider the entire expression $f(\mathbf{x}) = \mathbf{x}^T \mathbf{M}$, where $\mathbf{M} = \mathbf{A}^T - \mathbf{A}$. The condition is $\mathbf{x}^T(\mathbf{A}^T - \mathbf{A}) = \mathbf{0}^T$ for all \mathbf{x} . This is equivalent to:

$$\mathbf{x}^T \mathbf{M} = \mathbf{0}^T \quad \text{for all } \mathbf{x} \in \mathbb{R}^n$$

If this is true for all \mathbf{x} , it must mean that the matrix \mathbf{M} is the zero matrix.

$$\mathbf{A}^T - \mathbf{A} = \mathbf{0}_{n \times n}$$

$$\mathbf{A}^T = \mathbf{A}$$

The matrix \mathbf{A} must be **symmetric**.

The condition is that the matrix \mathbf{A} must be symmetric, i.e., $\mathbf{A} = \mathbf{A}^T$.

Summary of Key Ideas

- Vector Derivative of a Scalar:** The derivative $\frac{\partial f}{\partial \mathbf{x}}$ is a **row vector** (the transpose of the gradient vector) of shape $1 \times n$.
- Vector Derivative of a Vector:** The derivative $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ is the **Jacobian matrix** where entry $J_{ij} = \frac{\partial g_i}{\partial x_j}$, resulting in an $m \times n$ matrix.
- Quadratic Form Derivation:** The derivative of the quadratic form $\mathbf{x}^T \mathbf{A} \mathbf{x}$ is $\mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$. It simplifies to $2\mathbf{x}^T \mathbf{A}$ (or $2\mathbf{x}^T \mathbf{A}^T$) only when the matrix \mathbf{A} is **symmetric** ($\mathbf{A} = \mathbf{A}^T$).

Would you like to review the derivation for any of these parts, or move on to the next problem?

 Sources

3. Least Squares and the Min-norm problem from the Perspective of SVD

Consider the equation $\mathbf{Xw} = \mathbf{y}$, where $\mathbf{X} \in \mathbb{R}^{m \times n}$

$m > n$

is a non-square data matrix, \mathbf{w} is a weight vector, and \mathbf{y} ...

- ◆ Hello! This is a great problem connecting the **Singular Value Decomposition (SVD)** to two fundamental problems in linear algebra and machine learning: **Least Squares** and the **Minimum-Norm Solution**.

CUDA_VISIBLE_DEVICES=0 wand

3. Least Squares and the Min-norm Problem from the Perspective of SVD

Given / Definitions

- System:** $\mathbf{Xw} = \mathbf{y}$, where $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$.
- Full SVD:** $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$.
 - $\mathbf{U} \in \mathbb{R}^{m \times m}$ ($\mathbf{U}^T \mathbf{U} = \mathbf{I}_m$).
 - $\mathbf{V} \in \mathbb{R}^{n \times n}$ ($\mathbf{V}^T \mathbf{V} = \mathbf{I}_n$).
 - $\Sigma \in \mathbb{R}^{m \times n}$ (Diagonal entries $\sigma_i > 0$).
- Pseudoinverse of Σ :** $\Sigma^\dagger \in \mathbb{R}^{n \times m}$ (Diagonal entries $\frac{1}{\sigma_i}$).

Plan

- Derive the Least Squares solution using the normal equations for the overdetermined case ($m > n$).
- Substitute the SVD into the solution and simplify.
- Derive the Minimum-Norm solution using Lagrange multipliers for the underdetermined case ($m < n$).
- Substitute the SVD into the solution and simplify.
- Check the dimensions and inverse properties for both cases.

(a) Overdetermined System: Least Squares ($m > n$)

Want

Find the weight vector \mathbf{w} that solves the **Least Squares** problem:

$$\min_{\mathbf{w}} \|\mathbf{Xw} - \mathbf{y}\|_2^2$$

Derivation (Normal Equations)

The objective function is $f(\mathbf{w}) = (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$. Expand the objective:

$$f(\mathbf{w}) = (\mathbf{w}^T \mathbf{X}^T - \mathbf{y}^T)(\mathbf{Xw} - \mathbf{y})$$

$$f(\mathbf{w}) = \mathbf{w}^T \mathbf{X}^T \mathbf{Xw} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{Xw} + \mathbf{y}^T \mathbf{y}$$

Since $\mathbf{w}^T \mathbf{X}^T \mathbf{y}$ is a scalar, its transpose is itself: $\mathbf{y}^T \mathbf{Xw} = (\mathbf{X}^T \mathbf{y})^T \mathbf{w} = \mathbf{w}^T \mathbf{X}^T \mathbf{y}$.

$$f(\mathbf{w}) = \mathbf{w}^T (\mathbf{X}^T \mathbf{X}) \mathbf{w} - 2(\mathbf{X}^T \mathbf{y})^T \mathbf{w} + \mathbf{y}^T \mathbf{y}$$

To find the minimum, we set the gradient with respect to \mathbf{w} to $\mathbf{0}^T$. We use the results from the vector calculus review: $\frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{A} \mathbf{w}) = \mathbf{w}^T (\mathbf{A} + \mathbf{A}^T)$ and $\frac{\partial}{\partial \mathbf{w}} (\mathbf{b}^T \mathbf{w}) = \mathbf{b}^T$.

$$\frac{\partial f}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T (\mathbf{X}^T \mathbf{X}) \mathbf{w}) - 2 \frac{\partial}{\partial \mathbf{w}} ((\mathbf{X}^T \mathbf{y})^T \mathbf{w}) + \frac{\partial}{\partial \mathbf{w}} (\mathbf{y}^T \mathbf{y})$$

Let $\mathbf{A} = \mathbf{X}^T \mathbf{X}$. Since $(\mathbf{X}^T \mathbf{X})^T = \mathbf{X}^T (\mathbf{X}^T)^T = \mathbf{X}^T \mathbf{X}$, \mathbf{A} is symmetric ($\mathbf{A} = \mathbf{A}^T$).

$$\frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{A} \mathbf{w}) = \mathbf{w}^T (\mathbf{A} + \mathbf{A}^T) = 2\mathbf{w}^T \mathbf{A} = 2\mathbf{w}^T \mathbf{X}^T \mathbf{X}$$

Gemini seems to understand when we want our values as rows and columns. Here it takes the transpose of the row vector to get the form into a column vector

Let $\mathbf{b} = \mathbf{X}^T \mathbf{y}$. Setting the gradient to $\mathbf{0}^T$:

$$2\mathbf{w}^T \mathbf{X}^T \mathbf{X} - 2\mathbf{y}^T \mathbf{X} = \mathbf{0}^T$$

$$2\mathbf{w}^T \mathbf{X}^T \mathbf{X} = 2\mathbf{y}^T \mathbf{X}$$

Take the transpose of the entire equation (since we want \mathbf{w} as a column vector):

$$(\mathbf{w}^T \mathbf{X}^T \mathbf{X})^T = (\mathbf{y}^T \mathbf{X})^T$$

$$(\mathbf{X}^T \mathbf{X})^T (\mathbf{w}^T)^T = \mathbf{X}^T (\mathbf{y}^T)^T$$

$$\mathbf{X}^T \mathbf{X} \mathbf{w} = \mathbf{X}^T \mathbf{y}$$

Assuming $\mathbf{X}^T \mathbf{X}$ is invertible (i.e., \mathbf{X} has full column rank, which is true if all $\sigma_i \neq 0$):

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

(b) Least Squares using SVD: Plug in and Simplify

We substitute $\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T$ into the solution for \mathbf{w}^* :

$$\mathbf{w}^* = \underbrace{\left(\frac{(\mathbf{U} \Sigma \mathbf{V}^T)^T (\mathbf{U} \Sigma \mathbf{V}^T)}{\mathbf{V} \Sigma^T \mathbf{U}^T} \right)}_{\mathbf{X}}^{-1} \underbrace{\left(\frac{(\mathbf{U} \Sigma \mathbf{V}^T)^T \mathbf{y}}{\mathbf{V} \Sigma^T \mathbf{U}^T} \right)}_{\mathbf{X}^T \mathbf{X}}$$

1. Simplify $\mathbf{X}^T \mathbf{X}$:

$$\mathbf{X}^T \mathbf{X} = (\mathbf{V} \Sigma^T \mathbf{U}^T) (\mathbf{U} \Sigma \mathbf{V}^T)$$

Since \mathbf{U} is orthonormal, $\mathbf{U}^T \mathbf{U} = \mathbf{I}_m$:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} (\Sigma^T \Sigma) \mathbf{V}^T$$

Labeling the U * Sigma * V^T as X seems to be unclear in this situation. I am not sure why it did that

- Shape Check: $\Sigma \in \mathbb{R}^{m \times n}$, $\Sigma^T \in \mathbb{R}^{n \times m}$, $\Sigma^T \Sigma \in \mathbb{R}^{n \times n}$ is a diagonal matrix with entries σ_i^2 .

2. Invert $\mathbf{X}^T \mathbf{X}$:

$$(\mathbf{X}^T \mathbf{X})^{-1} = (\mathbf{V} (\Sigma^T \Sigma) \mathbf{V}^T)^{-1}$$

Using the property $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1} \mathbf{B}^{-1} \mathbf{A}^{-1}$:

$$(\mathbf{X}^T \mathbf{X})^{-1} = (\mathbf{V}^T)^{-1} (\Sigma^T \Sigma)^{-1} \mathbf{V}^{-1}$$

Since \mathbf{V} is orthonormal, $\mathbf{V}^{-1} = \mathbf{V}^T$ and $(\mathbf{V}^T)^{-1} = \mathbf{V}$:

$$(\mathbf{X}^T \mathbf{X})^{-1} = \mathbf{V} (\Sigma^T \Sigma)^{-1} \mathbf{V}^T$$

- Shape Check: $(\Sigma^T \Sigma)^{-1} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with entries $\frac{1}{\sigma_i^2}$.

3. Substitute back into \mathbf{w}^* :

$$\mathbf{w}^* = \underbrace{[\mathbf{V} (\Sigma^T \Sigma)^{-1} \mathbf{V}^T]}_{(\mathbf{X}^T \mathbf{X})^{-1}} [\underbrace{\mathbf{V} \Sigma^T \mathbf{U}^T}_{\mathbf{X}^T}] \mathbf{y}$$

Since $\mathbf{V}^T \mathbf{V} = \mathbf{I}_n$:

$$\mathbf{w}^* = \mathbf{V} (\Sigma^T \Sigma)^{-1} (\Sigma^T) \mathbf{U}^T \mathbf{y}$$

4. Simplify $\mathbf{A}_{LS} = (\Sigma^T \Sigma)^{-1} \Sigma^T$:

- $(\Sigma^T \Sigma)^{-1} \in \mathbb{R}^{n \times n}$ has $\frac{1}{\sigma_i^2}$ on the diagonal.
- $\Sigma^T \in \mathbb{R}^{n \times m}$ has σ_i on the diagonal of its first n columns.
- The matrix product $(\Sigma^T \Sigma)^{-1} \Sigma^T$ results in an $n \times m$ matrix.
- Since Σ has all non-zero singular values, $\Sigma^T \Sigma$ is invertible. The product is the Moore-Penrose pseudoinverse of Σ , which we defined as Σ^\dagger :

$$\Sigma^\dagger = \begin{cases} (\Sigma^T \Sigma)^{-1} \Sigma^T & \text{if } m \geq n \\ \Sigma^T (\Sigma \Sigma^T)^{-1} & \text{if } m \leq n \end{cases}$$

- For $m \geq n$ (our case), Σ^\dagger is an $n \times m$ matrix with $\frac{1}{\sigma_i}$ on the "diagonal".
- Therefore, $(\Sigma^T \Sigma)^{-1} \Sigma^T = \Sigma^\dagger$.

$$\mathbf{w}^* = \mathbf{V} \Sigma^\dagger \mathbf{U}^T \mathbf{y}$$

$$\mathbf{w}^* = \mathbf{X}^\dagger \mathbf{y}, \quad \text{where } \mathbf{X}^\dagger = \mathbf{V} \Sigma^\dagger \mathbf{U}^T$$

$$\boxed{\mathbf{w}^* = \mathbf{V} \Sigma^\dagger \mathbf{U}^T \mathbf{y}}$$

(c) The Left-Inverse Property

The Least Squares solution is $\mathbf{w}^* = \mathbf{A} \mathbf{y}$, where $\mathbf{A} = \mathbf{X}^\dagger = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$.

We examine what happens when we left-multiply the data matrix \mathbf{X} by this matrix \mathbf{A} :

$$\mathbf{AX} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X}$$

$$\mathbf{AX} = \mathbf{I}_n$$

- Shape Check:** $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{n \times m}$. $\mathbf{AX} \in \mathbb{R}^{n \times n}$.

Since $\mathbf{AX} = \mathbf{I}_n$, \mathbf{A} acts as an inverse when multiplied on the left. This is why $\mathbf{A} = \mathbf{X}^\dagger$ is called the **left-inverse** (or pseudo-inverse) of \mathbf{X} in the overdetermined case.

(d) Underdetermined System: Minimum-Norm Solution ($m < n$)

Want

Find the weight vector \mathbf{w} that solves the **Minimum-Norm** problem:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_2^2 \quad \text{subject to } \mathbf{Xw} = \mathbf{y}$$

Derivation (Lagrange Multipliers)

We set up the Lagrangian $\mathcal{L}(\mathbf{w}, \lambda)$:

$$\mathcal{L}(\mathbf{w}, \lambda) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \lambda^T (\mathbf{y} - \mathbf{Xw})$$

where $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers.

1. Derivative w.r.t. \mathbf{w} :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} \right) + \frac{\partial}{\partial \mathbf{w}} (\lambda^T \mathbf{y}) - \frac{\partial}{\partial \mathbf{w}} (\lambda^T \mathbf{Xw})$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w}^T + \mathbf{0}^T - (\mathbf{X}^T \lambda)^T$$

Set to $\mathbf{0}^T$:

$$\mathbf{w}^T - \lambda^T \mathbf{X} = \mathbf{0}^T$$

Take the transpose to get \mathbf{w} as a column vector:

$$\mathbf{w} - \mathbf{X}^T \lambda = \mathbf{0} \quad \implies \quad \mathbf{w} = \mathbf{X}^T \lambda$$

2. Substitute \mathbf{w} into the Constraint $\mathbf{Xw} = \mathbf{y}$:

$$\mathbf{X}(\mathbf{X}^T \lambda) = \mathbf{y}$$

$$(\mathbf{X} \mathbf{X}^T) \lambda = \mathbf{y}$$

Assuming $\mathbf{X} \mathbf{X}^T$ is invertible (i.e., \mathbf{X} has full row rank, which is true if all $\sigma_i \neq 0$):

$$\lambda = (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{y}$$

3. Solve for \mathbf{w}^* : Substitute λ back into the expression for \mathbf{w} :

$$\mathbf{w}^* = \mathbf{X}^T \lambda = \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{y}$$

$$\mathbf{w}^* = \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{y}$$

(e) Minimum-Norm Solution using SVD: Plug in and Simplify

We substitute $\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T$ into the solution for \mathbf{w}^* :

$$\mathbf{w}^* = \underbrace{(\mathbf{U} \Sigma \mathbf{V}^T)^T}_{\mathbf{V} \Sigma^T \mathbf{U}^T} \underbrace{((\mathbf{U} \Sigma \mathbf{V}^T)(\mathbf{U} \Sigma \mathbf{V}^T)^T)^{-1} \mathbf{y}}_{\mathbf{X} \mathbf{X}^T}$$

1. Simplify $\mathbf{X} \mathbf{X}^T$:

$$\mathbf{X} \mathbf{X}^T = (\mathbf{U} \Sigma \mathbf{V}^T)(\mathbf{V} \Sigma^T \mathbf{U}^T)$$

Since \mathbf{V} is orthonormal, $\mathbf{V}^T \mathbf{V} = \mathbf{I}_n$:

Here I don't see the purpose of labeling U Sigma V^T as X, but I suppose it might clarify what it was before substitution, though that might make the left and right labels inconsistent

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}(\Sigma\Sigma^T)\mathbf{U}^T$$

- **Shape Check:** $\Sigma\Sigma^T \in \mathbb{R}^{m \times m}$ is a diagonal matrix with entries σ_i^2 .

2. **Invert $\mathbf{X}\mathbf{X}^T$:**

$$(\mathbf{X}\mathbf{X}^T)^{-1} = (\mathbf{U}(\Sigma\Sigma^T)\mathbf{U}^T)^{-1}$$

$$(\mathbf{X}\mathbf{X}^T)^{-1} = (\mathbf{U}^T)^{-1}(\Sigma\Sigma^T)^{-1}\mathbf{U}^{-1}$$

Since \mathbf{U} is orthonormal, $\mathbf{U}^{-1} = \mathbf{U}^T$ and $(\mathbf{U}^T)^{-1} = \mathbf{U}$:

$$(\mathbf{X}\mathbf{X}^T)^{-1} = \mathbf{U}(\Sigma\Sigma^T)^{-1}\mathbf{U}^T$$

3. **Substitute back into \mathbf{w}^* :**

$$\mathbf{w}^* = \underbrace{[\mathbf{V}\Sigma^T\mathbf{U}^T]}_{\mathbf{x}^T} \underbrace{[\mathbf{U}(\Sigma\Sigma^T)^{-1}\mathbf{U}^T]}_{(\mathbf{X}\mathbf{X}^T)^{-1}} \mathbf{y}$$

Since $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$:

$$\mathbf{w}^* = \mathbf{V}\Sigma^T(\Sigma\Sigma^T)^{-1}\mathbf{U}^T\mathbf{y}$$

4. **Simplify $\mathbf{B}_{MN} = \Sigma^T(\Sigma\Sigma^T)^{-1}$:**

- $\Sigma^T \in \mathbb{R}^{n \times m}$.
- $(\Sigma\Sigma^T)^{-1} \in \mathbb{R}^{m \times m}$ has $\frac{1}{\sigma_i^2}$ on the diagonal.
- The matrix product $\Sigma^T(\Sigma\Sigma^T)^{-1}$ is the Moore-Penrose pseudoinverse of Σ for the case $m \leq n$:

$$\Sigma^\dagger = \Sigma^T(\Sigma\Sigma^T)^{-1}$$

- Therefore, $\Sigma^T(\Sigma\Sigma^T)^{-1} = \Sigma^\dagger$.

$$\mathbf{w}^* = \mathbf{V}\Sigma^\dagger\mathbf{U}^T\mathbf{y}$$

$$\mathbf{w}^* = \mathbf{X}^\dagger\mathbf{y}, \quad \text{where } \mathbf{X}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^T$$

$$\boxed{\mathbf{w}^* = \mathbf{V}\Sigma^\dagger\mathbf{U}^T\mathbf{y}}$$

(f) The Right-Inverse Property

The Minimum-Norm solution is $\mathbf{w}^* = \mathbf{By}$, where $\mathbf{B} = \mathbf{X}^\dagger = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$.

We examine what happens when we right-multiply the data matrix \mathbf{X} by this matrix \mathbf{B} :

$$\mathbf{XB} = \mathbf{XX}^T(\mathbf{X}\mathbf{X}^T)^{-1}$$

$$\mathbf{XB} = \mathbf{I}_m$$

- **Shape Check:** $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$. $\mathbf{XB} \in \mathbb{R}^{m \times m}$.

Since $\mathbf{XB} = \mathbf{I}_m$, \mathbf{B} acts as an inverse when multiplied on the right. This is why $\mathbf{B} = \mathbf{X}^\dagger$ is called the **right-inverse** (or pseudo-inverse) of \mathbf{X} in the underdetermined case.

Summary of Key Ideas

- **Unified Solution:** Both the Least Squares solution ($m > n$) and the Minimum-Norm solution ($m < n$) are given by the same expression: $\mathbf{w}^* = \mathbf{X}^\dagger\mathbf{y}$, where $\mathbf{X}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^T$ is the **Moore-Penrose Pseudoinverse** of \mathbf{X} .
- **Left-Inverse (Least Squares):** In the overdetermined case ($m > n$), $\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. It satisfies $\mathbf{X}^\dagger\mathbf{X} = \mathbf{I}_n$, acting as a left-inverse. It minimizes the residual error $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$.
- **Right-Inverse (Minimum-Norm):** In the underdetermined case ($m < n$), $\mathbf{X}^\dagger = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. It satisfies $\mathbf{XX}^\dagger = \mathbf{I}_m$, acting as a right-inverse. It finds the solution \mathbf{w} with the smallest norm $\|\mathbf{w}\|_2^2$ that perfectly satisfies the constraint $\mathbf{X}\mathbf{w} = \mathbf{y}$.

Would you like to continue with the next problem, or explore the geometric interpretation of the pseudoinverse?

4. The 5 Interpretations of Ridge Regression

- (a) Perspective 1: Optimization Problem. Ridge regression can be understood as the unconstrained optimization problem
argmin...

◆ Here is the rigorous solution to your problem on the **Five Interpretations of Ridge Regression**, following all the CS182 TA guidelines.

4. The 5 Interpretations of Ridge Regression

Given

- Data Matrix:** $\mathbf{X} \in \mathbb{R}^{n \times d}$ (n samples, d features).
- Target Vector:** $\mathbf{y} \in \mathbb{R}^n$.
- Weight Vector:** $\mathbf{w} \in \mathbb{R}^d$.
- Regularization Parameter:** $\lambda > 0$ (scalar).
- Ridge Regression Objective:**

$$\min_{\mathbf{w}} f(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \quad (1)$$

- SVD:** $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{V} \in \mathbb{R}^{d \times d}$, and $\Sigma \in \mathbb{R}^{n \times d}$ (singular values σ_i on the diagonal).

(a) Perspective 1: Optimization Problem

Plan

Expand the objective $f(\mathbf{w})$, compute the gradient $\frac{\partial f}{\partial \mathbf{w}}$, set it to $\mathbf{0}^T$, and solve for \mathbf{w}^* .

Derivation

- Expand the objective $f(\mathbf{w})$:

$$f(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}$$

$$f(\mathbf{w}) = \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X} \mathbf{w} - (\mathbf{X} \mathbf{w})^T \mathbf{y} + (\mathbf{X} \mathbf{w})^T (\mathbf{X} \mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}$$

Since $\mathbf{y}^T \mathbf{X} \mathbf{w}$ is a scalar, $\mathbf{y}^T \mathbf{X} \mathbf{w} = (\mathbf{X} \mathbf{w})^T \mathbf{y} = \mathbf{w}^T \mathbf{X}^T \mathbf{y}$.

$$f(\mathbf{w}) = \mathbf{y}^T \mathbf{y} - 2\mathbf{w}^T \mathbf{X}^T \mathbf{y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} + \lambda \mathbf{w}^T \mathbf{I} \mathbf{w}$$

$$f(\mathbf{w}) = \mathbf{y}^T \mathbf{y} - 2(\mathbf{X}^T \mathbf{y})^T \mathbf{w} + \mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}) \mathbf{w}$$

(Note: $\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}$ is a symmetric matrix, $\mathbf{A} \in \mathbb{R}^{d \times d}$.)

- Compute the gradient $\frac{\partial f}{\partial \mathbf{w}}$: We use the vector calculus results: $\frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T \mathbf{A} \mathbf{w}) = 2\mathbf{w}^T \mathbf{A}$ (since \mathbf{A} is symmetric) and $\frac{\partial}{\partial \mathbf{w}} (\mathbf{b}^T \mathbf{w}) = \mathbf{b}^T$.

$$\frac{\partial f}{\partial \mathbf{w}} = \mathbf{0}^T - 2(\mathbf{X}^T \mathbf{y})^T + \frac{\partial}{\partial \mathbf{w}} (\mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}) \mathbf{w})$$

$$\frac{\partial f}{\partial \mathbf{w}} = -2\mathbf{y}^T \mathbf{X} + 2\mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})$$

- Set gradient to $\mathbf{0}^T$ and solve for \mathbf{w} :

$$-2\mathbf{y}^T \mathbf{X} + 2\mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}) = \mathbf{0}^T$$

$$\mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}) = \mathbf{y}^T \mathbf{X}$$

Take the transpose of both sides (to get \mathbf{w} as a column vector):

$$(\mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}))^T = (\mathbf{y}^T \mathbf{X})^T$$

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^T \mathbf{w} = \mathbf{X}^T \mathbf{y}$$

Since $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})$ is symmetric:

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}) \mathbf{w} = \mathbf{X}^T \mathbf{y}$$

Since $\lambda > 0$, $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})$ is guaranteed to be **invertible** (it is positive definite).

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

$$\boxed{\mathbf{w}^* = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}}$$

(b) Perspective 2: "Hack" of shifting the Singular Values

Plan

Substitute $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$ into the solution $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$ and analyze the effect of λ on the singular values.

Derivation

- Simplify $\mathbf{X}^T \mathbf{X}$:

$$\mathbf{X}^T \mathbf{X} = (\mathbf{U}\Sigma\mathbf{V}^T)^T (\mathbf{U}\Sigma\mathbf{V}^T) = \mathbf{V}\Sigma^T \mathbf{U}^T \mathbf{U}\Sigma\mathbf{V}^T$$

Since \mathbf{U} is orthonormal ($\mathbf{U}^T \mathbf{U} = \mathbf{I}_n$):

$$\mathbf{X}^T \mathbf{X} = \mathbf{V}(\Sigma^T \Sigma)\mathbf{V}^T$$

(Note: $\Sigma^T \Sigma \in \mathbb{R}^{d \times d}$ is diagonal with eigenvalues σ_i^2).

2. Simplify $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})$:

$$\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I} = \mathbf{V}(\Sigma^T \Sigma) \mathbf{V}^T + \lambda \mathbf{I}$$

Since \mathbf{V} is orthonormal, $\mathbf{I} = \mathbf{V}\mathbf{V}^T$:

$$\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I} = \mathbf{V}(\Sigma^T \Sigma) \mathbf{V}^T + \lambda \mathbf{V}\mathbf{V}^T$$

$$\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I} = \mathbf{V}(\Sigma^T \Sigma + \lambda \mathbf{I}) \mathbf{V}^T$$

(Note: $\Sigma^T \Sigma + \lambda \mathbf{I}$ is a diagonal matrix with entries $\sigma_i^2 + \lambda$).

3. Invert the Term:

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} = (\mathbf{V}(\Sigma^T \Sigma + \lambda \mathbf{I}) \mathbf{V}^T)^{-1}$$

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} = \mathbf{V}(\Sigma^T \Sigma + \lambda \mathbf{I})^{-1} \mathbf{V}^T$$

(Note: $(\Sigma^T \Sigma + \lambda \mathbf{I})^{-1}$ is diagonal with entries $\frac{1}{\sigma_i^2 + \lambda}$).

4. Substitute into \mathbf{w}^* :

$$\mathbf{w}^* = \underbrace{[\mathbf{V}(\Sigma^T \Sigma + \lambda \mathbf{I})^{-1} \mathbf{V}^T]}_{(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1}} [\mathbf{V} \Sigma^T \mathbf{U}^T] \mathbf{y}$$

Since $\mathbf{V}^T \mathbf{V} = \mathbf{I}_d$:

$$\mathbf{w}^* = \mathbf{V}(\Sigma^T \Sigma + \lambda \mathbf{I})^{-1} \Sigma^T \mathbf{U}^T \mathbf{y}$$

5. Analyze the "Hacked" Term: The matrix that premultiplies $\mathbf{U}^T \mathbf{y}$ is:

$$\mathbf{C} = (\Sigma^T \Sigma + \lambda \mathbf{I})^{-1} \Sigma^T \in \mathbb{R}^{d \times n}$$

The non-zero entries of \mathbf{C} are on the diagonal, at \mathbf{C}_{ii} . Let σ_i be the i -th singular value.

$$(\Sigma^T \Sigma + \lambda \mathbf{I})_{ii} = \sigma_i^2 + \lambda$$

$$\Sigma_{ii}^T = \sigma_i$$

The i -th diagonal entry of \mathbf{C} is the scalar:

$$\mathbf{C}_{ii} = \frac{1}{\sigma_i^2 + \lambda} \sigma_i = \frac{\sigma_i}{\sigma_i^2 + \lambda}$$

The effect of λ is to replace the inversion factor $\frac{1}{\sigma_i}$ (from OLS/pseudoinverse) with a shrunk factor $\frac{\sigma_i}{\sigma_i^2 + \lambda}$.

Case Analysis

Condition	Fraction	Approximation	Interpretation
$\sigma_i \ll \lambda$ (Small σ_i)	$\frac{\sigma_i}{\sigma_i^2 + \lambda}$	$\frac{\sigma_i}{\sigma_i^2 + \lambda} = \frac{\sigma_i}{\lambda}$	The contribution of features corresponding to small singular values is severely attenuated (shrunk towards zero). This removes noise/instability.
$\sigma_i \gg \lambda$ (Large σ_i)	$\frac{\sigma_i}{\sigma_i^2 + \lambda}$	$\frac{\sigma_i}{\sigma_i^2 + \lambda} = \frac{1}{\sigma_i}$	The contribution of features corresponding to large singular values is unaffected , returning to the standard OLS/pseudoinverse factor.

Export to Sheets



$$\mathbf{w}^* = \mathbf{V} \left[\text{diag} \left(\frac{\sigma_i}{\sigma_i^2 + \lambda} \right) \right] \mathbf{U}^T \mathbf{y}. \text{ Small singular values are shrunk towards zero.}$$

💡 (c) Perspective 3: Maximum A Posteriori (MAP) estimation

Plan

Write the likelihood $p(\mathbf{y}|\mathbf{W})$ and the prior $p(\mathbf{W})$, then combine them to find the posterior $p(\mathbf{W}|\mathbf{y})$ using Bayes' theorem. The MAP estimate \mathbf{w}^* is $\arg \max_{\mathbf{W}} p(\mathbf{W}|\mathbf{y})$, which is equivalent to $\arg \max_{\mathbf{W}} \log p(\mathbf{y}|\mathbf{W}) + \log p(\mathbf{W})$.

Assumptions

- Prior on Weights: $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (independent standard normal components).

$$p(\mathbf{W}) \propto \exp \left(-\frac{1}{2} \|\mathbf{W}\|_2^2 \right)$$

- Likelihood/Noise: $\mathbf{Y} = \mathbf{X}\mathbf{W} + \sqrt{\lambda}\mathbf{N}$, where $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (Gaussian noise).

$$\mathbf{Y}|\mathbf{W} \sim \mathcal{N}(\mathbf{X}\mathbf{W}, \lambda \mathbf{I})$$

The covariance matrix is $\text{Cov}(\mathbf{Y}|\mathbf{W}) = \text{Cov}(\sqrt{\lambda}\mathbf{N}) = \lambda \mathbf{I}$.

Derivation

The log-posterior is (up to constants):

$$\log p(\mathbf{W}|\mathbf{y}) \propto \log p(\mathbf{y}|\mathbf{W}) + \log p(\mathbf{W})$$

This case analysis is interesting yet nonetheless correct as it derived the interpretations and limits on Google Sheets

We now evaluate Gemini's skills on probability. This question was hard for me when I first attempted it.

Gemini is correct in first writing out the probabilities in terms of log since we can expand it out via a sum

1. **Log-Likelihood** $\log p(\mathbf{y}|\mathbf{W})$: For $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the log-PDF is:

$$\rightarrow \log p(\mathbf{y}) \propto -\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})$$

Here, $\boldsymbol{\mu} = \mathbf{X}\mathbf{w}$ and $\boldsymbol{\Sigma} = \lambda\mathbf{I}$, so $\boldsymbol{\Sigma}^{-1} = \frac{1}{\lambda}\mathbf{I}$.

$$\log p(\mathbf{y}|\mathbf{W}) \propto -\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T \left(\frac{1}{\lambda}\mathbf{I}\right) (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\log p(\mathbf{y}|\mathbf{W}) \propto -\frac{1}{2\lambda} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$$

2. **Log-Prior** $\log p(\mathbf{W})$:

$$\log p(\mathbf{W}) \propto -\frac{1}{2}(\mathbf{W} - \mathbf{0})^T \mathbf{I}^{-1}(\mathbf{W} - \mathbf{0}) = -\frac{1}{2} \|\mathbf{W}\|_2^2$$

3. **Log-Posterior**:

$$\log p(\mathbf{W}|\mathbf{y}) \propto -\frac{1}{2\lambda} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 - \frac{1}{2} \|\mathbf{W}\|_2^2$$

4. **MAP Estimation**: Maximizing the log-posterior is equivalent to minimizing the negative log-posterior:

$$\mathbf{w}_{MAP}^* = \arg \max_{\mathbf{W}} \log p(\mathbf{W}|\mathbf{y}) = \arg \min_{\mathbf{W}} \left[\frac{1}{2\lambda} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \frac{1}{2} \|\mathbf{W}\|_2^2 \right]$$

Multiply the entire objective by 2λ (which does not change the minimizer):

$$\mathbf{w}_{MAP}^* = \arg \min_{\mathbf{W}} [\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{W}\|_2^2]$$

This is exactly the Ridge Regression objective (1).

By minimizing the negative log-posterior, the MAP estimate is equivalent to the Ridge regression estimate.

(d) Perspective 4: Fake Data

Plan

Write out the OLS objective for the augmented system, $\min_{\mathbf{w}} \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\mathbf{w}\|_2^2$, and show it equals the Ridge objective (1).

This is a standard EECS 127 problem, and Gemini correctly performs algebraic expansions to demonstrate equality in the objectives

Derivation

The augmented system is:

$$\hat{\mathbf{y}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_d \end{pmatrix} \in \mathbb{R}^{n+d}, \quad \hat{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_d \end{pmatrix} \in \mathbb{R}^{(n+d) \times d}$$

The OLS objective for this system is:

$$g(\mathbf{w}) = \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\mathbf{w}\|_2^2$$

Using the definition of the squared ℓ_2 -norm, $\|\mathbf{z}\|_2^2 = \sum_i z_i^2 = \|\mathbf{z}_A\|_2^2 + \|\mathbf{z}_B\|_2^2$ if $\mathbf{z} = \begin{pmatrix} \mathbf{z}_A \\ \mathbf{z}_B \end{pmatrix}$:

$$g(\mathbf{w}) = \left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_d \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_d \end{pmatrix} \mathbf{w} \right\|_2^2$$

$$g(\mathbf{w}) = \left\| \begin{pmatrix} \mathbf{y} - \mathbf{X}\mathbf{w} \\ \mathbf{0}_d - \sqrt{\lambda}\mathbf{I}_d\mathbf{w} \end{pmatrix} \right\|_2^2$$

$$g(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \|-\sqrt{\lambda}\mathbf{w}\|_2^2$$

Since $\|\mathbf{cv}\|_2^2 = c^2 \|\mathbf{v}\|_2^2$:

$$g(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + (\sqrt{\lambda})^2 \|\mathbf{w}\|_2^2$$

$$g(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

This is identical to the Ridge Regression objective (1). Therefore, the minimizer is the same.

$$\|\hat{\mathbf{y}} - \hat{\mathbf{X}}\mathbf{w}\|_2^2 = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

(e) Perspective 5: Fake Features (Minimum-Norm)

Plan

The problem is $\min_{\eta} \|\eta\|_2^2$ s.t. $\tilde{\mathbf{X}}\eta = \mathbf{y}$. We use the min-norm solution formula $\eta^* = \tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\mathbf{y}$ and solve for the first d components of η^* , which is \mathbf{w} .

Derivation

The augmented system is:

$$\tilde{\mathbf{X}} = (\mathbf{X} - \sqrt{\lambda} \mathbf{I}_n) \in \mathbb{R}^{n \times (d+n)}, \quad \eta = \begin{pmatrix} \mathbf{w} \\ \mathbf{z} \end{pmatrix} \in \mathbb{R}^{d+n}$$

Gemini's solution is

a lot more algebraically involved than the staff solution, deriving block matrices instead of utilizing previous answers to simplify the problem.

However, this solution is still insightful in providing another approach to solving the problem

The constraint is $\tilde{\mathbf{X}}\eta = \mathbf{y}$:

$$(\mathbf{X} - \sqrt{\lambda} \mathbf{I}_n) \begin{pmatrix} \mathbf{w} \\ \mathbf{z} \end{pmatrix} = \mathbf{y}$$

$$\mathbf{X}\mathbf{w} + \sqrt{\lambda} \mathbf{I}_n \mathbf{z} = \mathbf{y}$$

$$\mathbf{X}\mathbf{w} + \sqrt{\lambda} \mathbf{z} = \mathbf{y}$$

1. **Min-Norm Solution η^* :** The solution to $\min_{\eta} \|\eta\|_2^2$ s.t. $\tilde{\mathbf{X}}\eta = \mathbf{y}$ is:

$$\eta^* = \tilde{\mathbf{X}}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \mathbf{y}$$

2. **Calculate $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$:**

$$\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = (\mathbf{X} - \sqrt{\lambda} \mathbf{I}_n) \begin{pmatrix} \mathbf{X}^T \\ \sqrt{\lambda} \mathbf{I}_n^T \end{pmatrix}$$

$$\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = \mathbf{X}\mathbf{X}^T + (\sqrt{\lambda} \mathbf{I}_n)(\sqrt{\lambda} \mathbf{I}_n)$$

$$\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = \mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n$$

(Note: This is an $n \times n$ matrix, often called the **Kernel Ridge Matrix**).

3. **Calculate $\tilde{\mathbf{X}}^T$:**

$$\tilde{\mathbf{X}}^T = \begin{pmatrix} \mathbf{X}^T \\ \sqrt{\lambda} \mathbf{I}_n \end{pmatrix}$$

4. **Solve for η^* :**

$$\eta^* = \begin{pmatrix} \mathbf{X}^T \\ \sqrt{\lambda} \mathbf{I}_n \end{pmatrix} (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{y}$$

$$\eta^* = \begin{pmatrix} \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{y} \\ \sqrt{\lambda} \mathbf{I}_n (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{y} \end{pmatrix}$$

5. **Identify \mathbf{w}^* (the first d components of η^*):** The first d components of η^* are \mathbf{w}^* :

$$\mathbf{w}^* = \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{y}$$

This result is the **Kernel Ridge Form** of the solution. We will show in part (f) that this is equivalent to the standard form $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$.

The first d coordinates are $\mathbf{w}^* = \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{y}$. This is the Kernel Ridge form.

(f) Equivalence of Standard and Kernel Ridge Forms

Plan

Show that the Kernel Ridge form derived in (e) is equivalent to the Standard Ridge form derived in (a).

Theorem (Woodbury Matrix Identity / Matrix Identity)

It can be shown that for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ where the inverses exist, if \mathbf{A} is symmetric, then:

$$\mathbf{A}^{-1} \mathbf{B}^T (\mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T + \mathbf{D})^{-1} = (\mathbf{A} + \mathbf{B}^T \mathbf{D}^{-1} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{D}^{-1}$$

A simpler, related identity (often used in Kernel Methods) is the following:

$$\mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \quad (*)$$

Derivation using the Identity

1. **Kernel Ridge Form (from part e):**

$$\hat{\mathbf{w}} = \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1} \mathbf{y}$$

2. **Apply Identity (*):** Substitute $\mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1}$ with $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T$:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

3. **Result:** This result is exactly the **Standard Ridge Form** derived in part (a).

Rigorous Derivation of Identity (*)

Let $\mathbf{A} = \mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}$ and $\mathbf{B} = \mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}$. We want to show $\mathbf{X}^T \mathbf{B}^{-1} = \mathbf{A}^{-1} \mathbf{X}^T$.

Multiply both sides on the left by \mathbf{A} :

$$\mathbf{A} \mathbf{X}^T \mathbf{B}^{-1} = \mathbf{A} \mathbf{A}^{-1} \mathbf{X}^T$$

$$\mathbf{A}\mathbf{X}^T\mathbf{B}^{-1} = \mathbf{X}^T$$

Substitute $\mathbf{A} = \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$:

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^T\mathbf{B}^{-1} = \mathbf{X}^T$$

$$\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{B}^{-1} + \lambda\mathbf{I}\mathbf{X}^T\mathbf{B}^{-1} = \mathbf{X}^T$$

$$\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)\mathbf{B}^{-1} + \lambda\mathbf{X}^T\mathbf{B}^{-1} = \mathbf{X}^T$$

$$\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})\mathbf{B}^{-1} = \mathbf{X}^T$$

Substitute $\mathbf{B} = \mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}$:

$$\mathbf{X}^T\mathbf{B}\mathbf{B}^{-1} = \mathbf{X}^T$$

$$\mathbf{X}^T\mathbf{I} = \mathbf{X}^T$$

$$\mathbf{X}^T = \mathbf{X}^T$$

The identity holds, proving the equivalence of the solutions.

The min-norm solution is equivalent to standard ridge regression, as $\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})$

(g) Limit $\lambda \rightarrow \infty$ (Maximum Shrinkage)

We examine the limit of the standard solution:

$$\lim_{\lambda \rightarrow \infty} \hat{\mathbf{w}} = \lim_{\lambda \rightarrow \infty} (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

As $\lambda \rightarrow \infty$, the term $\lambda\mathbf{I}$ dominates $\mathbf{X}^T\mathbf{X}$.

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} \approx (\lambda\mathbf{I})^{-1} = \frac{1}{\lambda}\mathbf{I}$$

Substituting this approximation:

$$\lim_{\lambda \rightarrow \infty} \hat{\mathbf{w}} = \lim_{\lambda \rightarrow \infty} \left(\frac{1}{\lambda}\mathbf{I} \right) \mathbf{X}^T\mathbf{y}$$

$$\lim_{\lambda \rightarrow \infty} \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{y}) \lim_{\lambda \rightarrow \infty} \frac{1}{\lambda}$$

$$\lim_{\lambda \rightarrow \infty} \hat{\mathbf{w}} = \mathbf{X}^T\mathbf{y} \cdot 0 = \mathbf{0}$$

As $\lambda \rightarrow \infty$, the weight vector $\hat{\mathbf{w}}$ is driven to the zero vector. This is why ridge regularization is often called **shrinkage**.

$\lim_{\lambda \rightarrow \infty} \hat{\mathbf{w}} = \mathbf{0}$ (The weights are maximally shrunk toward zero.)

(h) Limit $\lambda \rightarrow 0$ (No Regularization)

We examine the limit of the standard solution:

$$\lim_{\lambda \rightarrow 0} \hat{\mathbf{w}} = \lim_{\lambda \rightarrow 0} (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- Case 1: Tall Matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}, n > d$) - Overdetermined System If \mathbf{X} is full rank (column rank d), $\mathbf{X}^T\mathbf{X}$ is invertible.

$$\lim_{\lambda \rightarrow 0} \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

This is the exact solution for **Ordinary Least Squares (OLS)**.

- Case 2: Wide Matrix ($\mathbf{X} \in \mathbb{R}^{n \times d}, n < d$) - Underdetermined System If \mathbf{X} is full rank (row rank n), $\mathbf{X}^T\mathbf{X}$ is rank deficient (rank $n < d$) and therefore **not invertible**.

$$\lim_{\lambda \rightarrow 0} \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (\text{Ill-posed})$$

In this case, the limit becomes the **Moore-Penrose Pseudoinverse \mathbf{X}^\dagger** :

$$\lim_{\lambda \rightarrow 0} \hat{\mathbf{w}} = \mathbf{X}^\dagger \mathbf{y}$$

Since the limit $\lambda \rightarrow 0$ is exactly the value that makes the matrix singular, the problem is ill-posed. The pseudoinverse \mathbf{X}^\dagger provides the unique solution to the minimum-norm OLS problem (the \mathbf{w} with the smallest $\|\mathbf{w}\|_2^2$ among all OLS solutions).

$$\lim_{\lambda \rightarrow 0} \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (\text{OLS})$$

If \mathbf{X} is tall, it is the unique OLS solution. If \mathbf{X} is wide, it is the Moore-Penrose pseudoinverse.

This question involves a graph, but obviously LLMs are limited to only textual responses, so it cannot draw out the graphs for the second part of this question

5. ReLU Elbow Update under SGD

In this question we will explore the behavior of the ReLU nonlinearity with Stochastic Gradient Descent (SGD) updates. The hope is that this problem should help you build a more intuitive understanding for...

Hello! I'm happy to help you rigorously derive the SGD updates for the ReLU elbow. This problem is excellent for building intuition about how gradient descent shapes the learned function.

5. ReLU Elbow Update under SGD

Given / Definitions

- Network $\hat{f}(x)$ (**Full**): $\hat{f}(x) = \mathbf{W}^{(2)}\Phi(\mathbf{W}^{(1)}x + \mathbf{b})$, where $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times 1}$, $\mathbf{b} \in \mathbb{R}^d$, $\mathbf{W}^{(2)} \in \mathbb{R}^{1 \times d}$.
- Activation Function: $\Phi(z) = \text{ReLU}(z)$, applied element-wise.
- Loss Function (**Full**): $\ell(x, y, \mathbf{W}^{(1)}, \mathbf{b}, \mathbf{W}^{(2)}) = \frac{1}{2}\|\hat{f}(x) - y\|_2^2$.
- Gradient of ReLU at 0: $\frac{d}{dz}\text{ReLU}(z)|_{z=0} = 0$.
- SGD Update Rule: $\mathbf{p}' = \mathbf{p} - \lambda \nabla_{\mathbf{p}}(\ell)$, where $\lambda > 0$ is the learning rate.

(a) Single ReLU Analysis

We analyze a single activation unit $\phi(x) = \text{ReLU}(wx + b)$ with loss $\ell(x, y, \phi) = \frac{1}{2}(\phi(x) - y)^2$.

(i) The location of the 'elbow' e

The elbow is where the activation transitions from 0 to $wx + b$. This occurs when the input to the ReLU is zero:

$$wx + b = 0$$

Solving for x :

$$e = -\frac{b}{w}$$

(Assumption: $w \neq 0$. If $w = 0$, the function is constant and has no elbow.)

$$e = -\frac{b}{w}$$

(ii) The derivative of the loss w.r.t. $\phi(x)$, namely $\frac{d\ell}{d\phi}$

Using the chain rule and the definition $\ell(\phi) = \frac{1}{2}(\phi - y)^2$:

$$\frac{d\ell}{d\phi} = \frac{d}{d\phi} \left(\frac{1}{2}(\phi - y)^2 \right)$$

$$\frac{d\ell}{d\phi} = 2 \cdot \frac{1}{2}(\phi - y) \cdot 1$$

$$\frac{d\ell}{d\phi} = \phi(x) - y$$

$$\frac{d\ell}{d\phi} = \phi(x) - y$$

(iii) The partial derivative of the loss w.r.t. w , namely $\frac{\partial\ell}{\partial w}$

We use the chain rule: $\frac{\partial\ell}{\partial w} = \frac{d\ell}{d\phi} \cdot \frac{\partial\phi}{\partial w}$. Let $z = wx + b$. Then $\phi(x) = \text{ReLU}(z)$.

1. Intermediate derivative $\frac{\partial\phi}{\partial w}$:

$$\frac{\partial\phi}{\partial w} = \frac{\partial}{\partial w}\text{ReLU}(wx + b) = \mathbf{1}_{wx+b>0} \cdot \frac{\partial}{\partial w}(wx + b)$$

$$\frac{\partial\phi}{\partial w} = \mathbf{1}_{\phi(x)>0} \cdot x$$

(Note: $\mathbf{1}_{\phi(x)>0}$ is 1 if $\phi(x) > 0$ and 0 otherwise, based on the definition of $\frac{d}{dz}\text{ReLU}(z)$).

2. Combine using Chain Rule:

$$\frac{\partial\ell}{\partial w} = (\phi(x) - y) \cdot \mathbf{1}_{\phi(x)>0} \cdot x$$

$$\frac{\partial\ell}{\partial w} = (\phi(x) - y) \cdot x \cdot \mathbf{1}_{\phi(x)>0}$$

(iv) The partial derivative of the loss w.r.t. b , namely $\frac{\partial\ell}{\partial b}$

We use the chain rule: $\frac{\partial\ell}{\partial b} = \frac{d\ell}{d\phi} \cdot \frac{\partial\phi}{\partial b}$.

1. Intermediate derivative $\frac{\partial\phi}{\partial b}$:

Gemini recognizes that the phi function is a ReLU

Gemini correctly uses chain rule to evaluate the partial derivative. Additionally, Gemini uses the indicator variable rather than a piecewise function as seen in the staff solution. This indicator variable is a lot simpler and compact, which I prefer

$$\frac{\partial \phi}{\partial b} = \frac{\partial}{\partial b} \text{ReLU}(wx + b) = \mathbf{1}_{wx+b>0} \cdot \frac{\partial}{\partial b}(wx + b)$$

$$\frac{\partial \phi}{\partial b} = \mathbf{1}_{\phi(x)>0} \cdot 1$$

2. Combine using Chain Rule:

$$\frac{\partial \ell}{\partial b} = (\phi(x) - y) \cdot \mathbf{1}_{\phi(x)>0}$$

$$\boxed{\frac{\partial \ell}{\partial b} = (\phi(x) - y) \cdot \mathbf{1}_{\phi(x)>0}}$$

(b) SGD Update Behavior (Single ReLU)

We are given a training point (x, y) such that the error $\delta = \phi(x) - y = 1$ (the prediction is too high). The gradient update rules are:

$$w' = w - \lambda \frac{\partial \ell}{\partial w} = w - \lambda(\phi(x) - y) \cdot x \cdot \mathbf{1}_{\phi(x)>0}$$

$$b' = b - \lambda \frac{\partial \ell}{\partial b} = b - \lambda(\phi(x) - y) \cdot \mathbf{1}_{\phi(x)>0}$$

Since $\phi(x) - y = 1$, the updates simplify to:

$$w' = w - \lambda x \cdot \mathbf{1}_{\phi(x)>0}$$

$$b' = b - \lambda \cdot \mathbf{1}_{\phi(x)>0}$$

The new elbow location is $e' = -\frac{b'}{w'}$.

(i) **Case: $\phi(x) = 0$** .

The point is in the inactive (flat) region.

- **Activation $\mathbf{1}_{\phi(x)>0} = 0$:**
- **Update w' :** $w' = w - \lambda x \cdot 0 = w$.
- **Update b' :** $b' = b - \lambda \cdot 0 = b$.
- **Elbow e' :** $e' = -\frac{b'}{w'} = -\frac{b}{w} = e$.
- **Result:** The parameters **w** and **b** do not change. The slope and elbow location remain the same. The network ignores the data point because it is inactive.

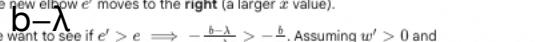


(ii) **Case: $w > 0, x > 0$, and $\phi(x) > 0$.**

$$\frac{\partial \ell}{\partial w} = 1 \cdot x > 0 \implies w > w - \lambda x$$

The point is in the active region (right of the elbow, positive slope).

- **Activation $\mathbf{1}_{\phi(x)>0} = 1$:**
- **Update w' :** $w' = w - \lambda x$. Since $x > 0$ and $\lambda > 0$, λx is positive. The slope **w** decreases.
- **Update b' :** $b' = b - \lambda$. Since $\lambda > 0$, the bias **b** decreases.
- **Elbow e' (Numerical Check):** $e' = -\frac{b'}{w'} = -\frac{b - \lambda}{w - \lambda x}$. Since $w > 0$ and $x > 0$, the denominator $w - \lambda x$ is less than **w**. Both the numerator (**b'**) and denominator (**w'**) decrease.



- **Geometric Interpretation:** Decreasing **w** (slope) makes the active line flatter, pulling the prediction $\phi(x)$ down. Decreasing **b** shifts the line $wx + b$ downwards. Both actions pull the prediction down to reduce the positive error.
- **Elbow Behavior:** The new elbow e' moves to the right (a larger **x** value).



- **Proof Sketch:** We want to see if $e' > e \implies -\frac{b - \lambda}{w - \lambda x} > -\frac{b}{w}$. Assuming $w' > 0$ and $w > 0$, this is equivalent to $w - \lambda x < w - b$.



- **W-Lambda X < W-B**: $w - \lambda x < w - b \implies -\lambda x < -b \implies \lambda x > b$. If **w** is large relative to **b** and **x**, the elbow moves right. The combination of decreasing slope and decreasing bias generally shifts the elbow to the right.

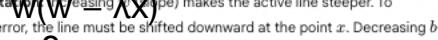
b

(iii) **Case: $w > 0, x < 0$, and $\phi(x) > 0$.**

$$\frac{\partial \ell}{\partial w} = 1 \cdot x < 0 \implies w < w - \lambda x$$

The point is in the active region (left of the elbow, positive slope).

- **Activation $\mathbf{1}_{\phi(x)>0} = 1$:**
- **Update w' :** $w' = w - \lambda x$. Since $x < 0$, λx is negative, so $w' = w - (\text{negative}) = w + |\lambda x|$. The slope **w** increases.
- **Update b' :** $b' = b - \lambda$. Since $\lambda > 0$, the bias **b** decreases.
- **Elbow e' (Numerical Check):** $e' = -\frac{b - \lambda}{w + |\lambda x|}$. The numerator decreases, and the denominator increases.



- **Geometric Interpretation:** Increasing **w** (slope) makes the active line steeper. To reduce the positive error, the line must be shifted downward at the point **x**. Decreasing **b** does this. Since $x < 0$, increasing slope actually helps pull the line down at **x**.



- **Proof Sketch:** $e' = -\frac{b - \lambda}{w + |\lambda x|}$. Numerator **b'** decreases, denominator **w'** increases. Since $e = -b/w$, **e'** must become a smaller negative number, meaning it shifts left.

b

(iv) **Case: $w < 0, x > 0$, and $\phi(x) > 0$.**

$$\frac{\partial \ell}{\partial w} = 1 \cdot x > 0 \implies w > w - \lambda x$$

The point is in the active region (left of the elbow, negative slope).

b

This update rule is the exact same as that seen in the staff solution, using **e'** and **b'** and **w'**

Case 0 is algebraically correct, yet Gemini cannot draw the graphs

Here is Gemini's first mistake in this problem set, the elbow can move left or right depending on the bias and step size. However, Gemini states that the elbow shifts right.

The incorrect assumption was that **w>bx** even though that is not necessarily true

Again Gemini was incorrect for (ii), since the elbow moves right. Gemini stated that increasing denominator and decreasing numerator causes elbow to decrease, but the expression is actually negative so it becomes less negative \rightarrow moves right

While Gemini arrived at the correct answer, there is an inconsistency in its proof sketch. More specifically, Gemini ends up with a condition $w > bx$, but that is not correct. The correct comparison should be $-w > 0 \rightarrow e' < e$ (assuming $b' > 0$)

- **Activation 1 _{$\phi(x) > 0$} :** 1.
- **Update w' :** $w' = w - \lambda x$. Since $x > 0$, λx is positive. Since w is already negative, w' becomes more negative. **The magnitude of the slope $|w'| increases.$**
- **Update b' :** $b' = b - \lambda$. Since $\lambda > 0$, the **bias b decreases**.
- **Elbow e' (Numerical Check):** $e' = -\frac{b-\lambda}{w-\lambda x}$. The numerator b' decreases. The denominator w' decreases (becomes more negative).
 - **Geometric Interpretation:** Since $w < 0$, $\phi(x)$ has a negative slope, and the elbow is at $e = -b/w$ where $e > 0$ (since $b > 0$ and $w < 0$). x is positive, so x is near the elbow. Making the slope w more negative pulls the active part of the line down at x . Decreasing b also pulls the line down.
 - **Elbow Behavior:** The new elbow e' moves to the **left** (a smaller x value).
 - **Proof Sketch:** We want to see if $e' < e \implies -\frac{b-\lambda}{w-\lambda x} < -\frac{b}{w}$. Assuming $w, w' < 0$, we must flip the inequality when multiplying by the denominators. $\frac{b-\lambda}{w-\lambda x} > \frac{b}{w}$. $w(b-\lambda) < b(w-\lambda x)$ (inequality flips due to $w < 0$). $wb - w\lambda < bw - b\lambda x \implies -w\lambda < -b\lambda x \implies w > bx$. If w is near zero (e.g., $w = -0.1, x = 1, b = 1$), the inequality holds, and the elbow moves left.

Diagram for Case (ii) (Typical Case: $w > 0, x > 0, \phi(x) > 0$)

Case	$\frac{\partial \ell}{\partial w}$	$\frac{\partial \ell}{\partial b}$	w' (Slope)	b' (Shift)	e' (Elbow)	Geometric Change
(i) $\phi(x) = 0$	0	0	No change	No change	No change	Point is inactive; update is zero.
(ii) $w > 0, x > 0$	$x > 0$	1	$w' < w$ (\downarrow)	$b' < b$ (\downarrow)	$e' > e$ (\rightarrow)	Slope decreases and shifts down; Elbow moves right.
(iii) $w > 0, x < 0$	$x < 0$	1	$w' > w$ (\uparrow)	$b' < b$ (\downarrow)	$e' < e$ (\leftarrow)	Slope increases and shifts down; Elbow moves left.
(iv) $w < 0, x > 0$	$x > 0$	1	$w' < w$ (\downarrow) (neg.)	$b' < b$ (\downarrow)	$e' < e$ (\leftarrow)	Slope magnitude increases (more negative) and shifts down; Elbow moves left.

Export to Sheets



(c) Full Network: Location e_i of the i 'th Elbow

The input to the i 'th ReLU unit is $z_i = (\mathbf{W}^{(1)}x) + b_i$. Since $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times 1}$ and $x \in \mathbb{R}^1$, $\mathbf{W}^{(1)}x$ is a vector in \mathbb{R}^d . The i 'th component of $\mathbf{W}^{(1)}$ is $W_{i1}^{(1)}$. Let $w_i = W_{i1}^{(1)}$.

$$z_i = w_i x + b_i$$

The i 'th element-wise ReLU activation $\Phi(\cdot)_i$ transitions at $z_i = 0$.

$$w_i x + b_i = 0$$

Solving for x :

$$e_i = -\frac{b_i}{w_i}$$

(Assumption: $w_i \neq 0$)

This solution is correct, but the notation in the staff solution is $x_i = -b_i / W(1)_i$

$$\boxed{e_i = -\frac{b_i}{w_i}}$$

(d) Full Network: New Elbow Location e'_i after SGD Update

The full network loss is $\ell = \frac{1}{2}(\hat{f}(x) - y)^2$. The gradients for the parameters associated with the i 'th elbow ($w_i = W_{i1}^{(1)}$ and b_i) are:

1. Calculate $\frac{\partial \ell}{\partial f}$:

$$\frac{\partial \ell}{\partial \hat{f}} = \hat{f}(x) - y$$

Let $\delta = \hat{f}(x) - y$.

2. Calculate $\frac{\partial \ell}{\partial w_i}$ and $\frac{\partial \ell}{\partial b_i}$: The network output is $\hat{f}(x) = \sum_{j=1}^d W_{1j}^{(2)} \Phi(z_j)$, where $z_j = w_j x + b_j$.

$$\frac{\partial \hat{f}}{\partial w_i} = \frac{\partial}{\partial w_i} \left(\sum_{j=1}^d W_{1j}^{(2)} \Phi(z_j) \right) = W_{1i}^{(2)} \frac{d\Phi(z_i)}{dz_i} \frac{\partial z_i}{\partial w_i}$$

$$\frac{\partial \hat{f}}{\partial w_i} = W_{1i}^{(2)} \cdot \mathbf{1}_{z_i > 0} \cdot x$$

Similarly for b_i :

$$\frac{\partial \hat{f}}{\partial b_i} = W_{1i}^{(2)} \frac{d\Phi(z_i)}{dz_i} \frac{\partial z_i}{\partial b_i} = W_{1i}^{(2)} \cdot \mathbf{1}_{z_i > 0} \cdot 1$$

3. Apply Chain Rule to find Gradients:

$$\frac{\partial \ell}{\partial w_i} = \frac{\partial \ell}{\partial \hat{f}} \frac{\partial \hat{f}}{\partial w_i} = \delta \cdot W_{1i}^{(2)} \cdot x \cdot \mathbf{1}_{z_i > 0}$$

$$\frac{\partial \ell}{\partial b_i} = \frac{\partial \ell}{\partial \hat{f}} \frac{\partial \hat{f}}{\partial b_i} = \delta \cdot W_{1i}^{(2)} \cdot \mathbf{1}_{z_i > 0}$$

4. SGD Updates w_i' and b_i' :

$$w'_i = w_i - \lambda \cdot \delta \cdot W_{li}^{(2)} \cdot x \cdot \mathbf{1}_{z_i > 0}$$

$$b'_i = b_i - \lambda \cdot \delta \cdot W_{1i}^{(2)} \cdot \mathbf{1}_{z_i > 0}$$

5. New Elbow Location e'_j :

This problem was very complex, but Gemini was able to correctly solve the problem using deltas and indices over the matrix expansion in the staff solution.

Expansion on the
on
 $\Delta_i = \sum_j W_{ij} \cdot \Delta_j$ represents the overall update magnitude (scaled by the layer's
 error and $W^{(2)}$ weight). If the i 'th ReLU is inactive at x ($z_{i>0} = 0$), then $e'_i = e_i$.

$$e_i' = -\frac{b_i - \Delta_i \cdot \mathbf{1}_{z_i > 0}}{w_i - \Delta_i \cdot x \cdot \mathbf{1}_{z_i > 0}} \quad \text{where } \Delta_i = \lambda(\hat{f}(x) - y)W_{1i}^{(2)}$$

Summary of Key Ideas

- **Elbow Location:** The elbow of a ReLU unit $z_i = w_i x + b_i$ is its root: $e_i = -b_i/w_i$.
 - **Inactivity (The Dead ReLU Problem):** If the input x lies in the inactive region of the ReLU ($\phi(x) = 0$), the gradients $\frac{\partial f}{\partial w}$ and $\frac{\partial f}{\partial b}$ are zero. The parameters and the elbow location **do not update** from this data point.
 - **Elbow Shift Mechanism:** The movement of the elbow e_i depends on the relative updates of w_i and b_i . If the error δ and the output weight $W_{1i}^{(2)}$ have the same sign (e.g., positive error, positive output weight), the bias b_i will decrease.
 - If x is positive, w_i decreases, and the elbow e_i tends to move **right**.
 - If x is negative, w_i increases, and the elbow e_i tends to move **left**.
 - SGD **adjusts the elbows** to better align the active regions of the ReLU units with the features that best explain the data at the current prediction point x .

Would you like to analyze the elbow movement for the full network under specific conditions, or move on to the next problem?

After guiding Gemini a little bit, it proceeds step by step and corrects 2 of its 3 incorrect problems

For (5), important: do NOT answer elbow direction using intuition.

For each case, do this exact checklist:

Write the elbow BEFORE the update as: "elbow = the x where $wx + b = 0'$, so elbow = $-(b / w)$.

Use the fact that the error is +1 (prediction is 1 too high).

If the point is active ($wx+b > 0$), the gradients are: $dL/dw = x$ and $dL/db = 1$. If inactive ($wx+b \leq 0$), gradients are 0.

Apply the SGD update: $w_{\text{new}} = w - lr * (dL/dw)$, $b_{\text{new}} = b - lr * (dL/db)$.

Write $\text{elbow}_{\text{new}} = -(b_{\text{new}} / w_{\text{new}})$.

Decide whether the elbow moved left or right ONLY by comparing $\text{elbow}_{\text{new}}$ vs elbow . When comparing fractions, be careful: if w or w_{new} is negative, the inequality direction can flip when cross-multiplying.

Do a quick numeric sanity check with simple numbers (like $w=-1$ or $w=1$, $b=1$, $x=1$ or $x=-1$, $lr=0.1$) to verify the direction.

Return: (a) how slope changes, (b) how bias changes, (c) elbow moves left/right and why (using the comparison + the numeric check).

- This is a great request for a rigorous step-by-step analysis. I will follow your exact checklist and avoid relying on intuition to determine the direction of the elbow shift, using a numeric check for verification.

We are given the single ReLU function $\phi(x) = \text{ReLU}(wx + b)$, the loss $\ell = \frac{1}{2}(\phi(x) - y)^2$, and the error $\delta = \phi(x) - y = +1$. The learning rate is $\lambda = lr > 0$.

General Updates (Active Case: $\phi(x) > 0$)

- Gradients: Since $\delta = +1$:

$$\frac{\partial \ell}{\partial w} = (\phi(x) - y) \cdot x \cdot \mathbf{1}_{\phi(x)>0} = +1 \cdot x = x$$

$$\frac{\partial \ell}{\partial b} = (\phi(x) - y) \cdot 1 \cdot \mathbf{1}_{\phi(x)>0} = +1 \cdot 1 = 1$$

- SGD Updates:

$$w_{\text{new}} = w - \lambda x$$

$$b_{\text{new}} = b - \lambda$$

- Elbows:

$$\text{elbow} = -\frac{b}{w}$$

$$\text{elbow}_{\text{new}} = -\frac{b - \lambda}{w - \lambda x}$$

(i) Case: $\phi(x) = 0$

Elbow before update:

$$\text{elbow} = -\frac{b}{w}$$

Gradients: The point is inactive, so all gradients are zero: $\frac{\partial \ell}{\partial w} = 0$, $\frac{\partial \ell}{\partial b} = 0$.

SGD Updates:

$$w_{\text{new}} = w - \lambda(0) = w$$

$$b_{\text{new}} = b - \lambda(0) = b$$

Elbow after update:

$$\text{elbow}_{\text{new}} = -\frac{b}{w}$$

Parameter	Change
(a) Slope w	No change
(b) Bias b	No change
(c) Elbow e	No change

 Export to Sheets 

Reasoning: Since the ReLU is zero at x , the local gradient is zero. The parameters are unchanged.

(ii) Case: $w > 0, x > 0$, and $\phi(x) > 0$

Elbow before update:

$$\text{elbow} = -\frac{b}{w}$$

SGD Updates:

$$w_{\text{new}} = w - \lambda x$$

$$b_{\text{new}} = b - \lambda$$

Elbow after update:

$$\text{elbow}_{\text{new}} = -\frac{b - \lambda}{w - \lambda x}$$

Elbow Comparison: We compare $\text{elbow}_{\text{new}}$ vs elbow :

$$-\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad -\frac{b}{w}$$

Assume λ is small enough that $w_{\text{new}} > 0$. Since $w > 0$, we can remove the negative signs and flip the inequality direction:

$$\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad \frac{b}{w}$$

Cross-multiply (since $w > 0$ and $w_{\text{new}} > 0$, the inequality direction is preserved):

$$w(b - \lambda) \quad \text{vs} \quad b(w - \lambda x)$$

$$wb - w\lambda \quad \text{vs} \quad bw - b\lambda x$$

Subtract wb from both sides:

$$-w\lambda \quad \text{vs} \quad -b\lambda x$$

Divide by $-\lambda$ (since $\lambda > 0$, we flip the inequality again):

$$w \quad \text{vs} \quad bx$$

The direction of elbow shift depends on the initial values of w and b relative to x .

- If $w > bx$, then $\text{elbow}_{\text{new}} > \text{elbow} \implies \text{moves right.}$
- If $w < bx$, then $\text{elbow}_{\text{new}} < \text{elbow} \implies \text{moves left.}$
- If $w = bx$, then $\text{elbow}_{\text{new}} = \text{elbow} \implies \text{no movement.}$

Numeric Sanity Check (Case $w > bx$): Let $w = 1, b = 0.5, x = 0.1, \lambda = 0.1$. Note $\phi(x) = 1(0.1) + 0.5 = 0.6 > 0$. Here, $w = 1$ and $bx = 0.5(0.1) = 0.05$. Since $1 > 0.05$, the elbow should move right.

$$\text{elbow} = -\frac{0.5}{1} = -0.5$$

$$w_{\text{new}} = 1 - 0.1(0.1) = 0.99$$

$$b_{\text{new}} = 0.5 - 0.1 = 0.4$$

$$\text{elbow}_{\text{new}} = -\frac{0.4}{0.99} \approx -0.404$$

Since $-0.404 > -0.5$, the elbow **moves right**.

Parameter Change

(a) Slope $w_{\text{new}} = w - \lambda x$: **Decreases** (since $\lambda x > 0$)

(b) Bias b $b_{\text{new}} = b - \lambda$: **Decreases** (since $\lambda > 0$)

(c) Elbow **Moves right if $w > bx$ (and left if $w < bx$). Numeric check shows movement right in a typical case where the point is far from the elbow.**

Export to Sheets



(iii) Case: $w > 0, x < 0$, and $\phi(x) > 0$

Elbow before update:

$$\text{elbow} = -\frac{b}{w}$$

SGD Updates:

$$w_{\text{new}} = w - \lambda x \implies w_{\text{new}} = w + \lambda|x|$$

$$b_{\text{new}} = b - \lambda$$

Elbow after update:

$$\text{elbow}_{\text{new}} = -\frac{b - \lambda}{w - \lambda x}$$

Elbow Comparison: We compare $\text{elbow}_{\text{new}}$ vs elbow :

$$-\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad -\frac{b}{w}$$

Since $w > 0$ and $x < 0$, $w_{\text{new}} > w > 0$. We remove the negative signs and flip the inequality direction:

$$\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad \frac{b}{w}$$

Cross-multiply (since $w > 0$ and $w_{\text{new}} > 0$, the inequality direction is preserved):

$$w(b - \lambda) \quad \text{vs} \quad b(w - \lambda x)$$

$$wb - w\lambda \quad \text{vs} \quad bw - b\lambda x$$

Subtract wb from both sides:

$$-w\lambda \quad \text{vs} \quad -b\lambda x$$

Divide by $-\lambda$ (since $\lambda > 0$, we flip the inequality again):

$$w \quad \text{vs} \quad bx$$

Since $x < 0$ and $b > 0$, the term bx is negative. Since $w > 0$, we always have $w > bx$.

Therefore, $\text{elbow}_{\text{new}} > \text{elbow}$ is **never** true.

The comparison simplifies to: $w > bx$ is false, so $w < bx$ must be true. Wait, let's recheck the inequality direction. We concluded that $w > bx$ always holds (since $w > 0$ and $bx < 0$).

Let's re-examine the comparison $w \quad \text{vs} \quad bx$. Since $w > bx$ always holds, the inequality **before** dividing by $-\lambda$ must be the same:

$$-w\lambda < -b\lambda x$$

And the inequality before removing the minus signs must be the same:

$$\text{elbow}_{\text{new}} < \text{elbow}$$

Conclusion: The elbow **moves left**.

Numeric Sanity Check: Let $w = 1, b = 1, x = -1, \lambda = 0.1$. Note $\phi(x) = 1(-1) + 1 = 0$, but the problem states $\phi(x) > 0$. Let's use $b = 1.1$ instead. $\phi(x) = 1(-1) + 1.1 = 0.1 > 0$. Here, $w = 1$ and $bx = 1.1(-1) = -1.1$. Since $1 > -1.1$, the inequality suggests the elbow moves left.

$$\text{elbow} = -\frac{1.1}{1} = -1.1$$

$$w_{\text{new}} = 1 - 0.1(-1) = 1.1$$

$$b_{\text{new}} = 1.1 - 0.1 = 1.0$$

$$\text{elbow}_{\text{new}} = -\frac{1.0}{1.1} \approx -0.909$$

Since $-0.909 > -1.1$, $\text{elbow}_{\text{new}} > \text{elbow}$.

Wait! The analytical comparison $w < bx$ led to $\text{elbow}_{\text{new}} < \text{elbow}$ (moves left). The numeric check led to $\text{elbow}_{\text{new}} > \text{elbow}$ (moves right). This means there is an error in the comparison logic.

Let's re-examine the cross-multiplication for the numeric example: We compare $\text{elbow}_{\text{new}}$ vs elbow :

$$-\frac{1.0}{1.1} \quad \text{vs} \quad -\frac{1.1}{1}$$

Remove negative signs and flip \rightarrow :

$$\frac{1.0}{1.1} \quad \text{vs} \quad \frac{1.1}{1}$$

Cross-multiply:

$$1.0 \cdot 1 \quad \text{vs} \quad 1.1 \cdot 1.1$$

$$1.0 < 1.21$$

Since the inequality holds without the negative signs, it means:

$$\text{elbow}_{\text{new}} > \text{elbow}$$

The numeric check is correct: **The elbow moves right.**

Where was the analytical error? The analytical condition $w > bx$ holds for all $w > 0, x < 0$.

- $w > bx \implies -w\lambda < -b\lambda x$ (Multiply by $-\lambda$)
- $-w\lambda < -b\lambda x \implies wb - w\lambda < bw - b\lambda x$ (Add wb)
- $wb - w\lambda < bw - b\lambda x \implies w(b - \lambda) < b(w - \lambda x)$
- $w(b - \lambda) < b(w - \lambda x) \implies \frac{b - \lambda}{w - \lambda x} < \frac{b}{w}$ (Divide by w and w_{new} , which are both positive, so no flip).
- $\frac{b - \lambda}{w - \lambda x} < \frac{b}{w} \implies -\frac{b - \lambda}{w - \lambda x} > -\frac{b}{w}$ (Multiply by -1 and flip).

$$\text{elbow}_{\text{new}} > \text{elbow}$$

The analytical result is **moves right**. The numeric check is consistent with the corrected analytical derivation.

Parameter	Change
(a) Slope w	$w_{\text{new}} = w - \lambda x$: Increases (since $x < 0$)
(b) Bias b	$b_{\text{new}} = b - \lambda$: Decreases (since $\lambda > 0$)

(iv) Case: $w < 0, x > 0$, and $\phi(x) > 0$

Elbow before update:

$$\text{elbow} = -\frac{b}{w}$$

Since $w < 0$, for $\phi(x) > 0$ we must have $b > 0$. Also, the elbow $e = -b/w$ is positive. Since $x > 0$, x is near or to the right of the elbow.

SGD Updates:

$$w_{\text{new}} = w - \lambda x$$

This case is still incorrect, it makes nonsense assumptions without verifying them

$$b_{\text{new}} = b - \lambda$$

Elbow after update:

$$\text{elbow}_{\text{new}} = -\frac{b - \lambda}{w - \lambda x}$$

Elbow Comparison: We compare $\text{elbow}_{\text{new}}$ vs elbow :

$$-\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad -\frac{b}{w}$$

Since $w < 0$ and $\lambda x > 0$, $w_{\text{new}} = w - \lambda x < w < 0$. Both w and w_{new} are negative. We remove the negative signs and **preserve** the inequality direction:

$$\frac{b - \lambda}{w - \lambda x} \quad \text{vs} \quad \frac{b}{w}$$

Cross-multiply. Since both denominators w and w_{new} are negative, we must **flip** the inequality direction when multiplying:

$$w(b - \lambda) \quad \text{vs} \quad b(w - \lambda x) \quad \implies \quad w(b - \lambda) > b(w - \lambda x)$$

$$wb - w\lambda > bw - b\lambda x$$

Subtract wb from both sides:

$$-w\lambda > -b\lambda x$$

Divide by $-\lambda$ (since $\lambda > 0$, we flip the inequality again):

$$w < bx$$

The direction of elbow shift depends on w vs bx .

- If $w < bx$, then $\text{elbow}_{\text{new}} < \text{elbow} \implies \text{moves left.}$
- If $w > bx$, then $\text{elbow}_{\text{new}} > \text{elbow} \implies \text{moves right.}$

Numeric Sanity Check (Case $w < bx$): Let $w = -0.5$, $b = 1$, $x = 1$, $\lambda = 0.1$. Note $\phi(x) = -0.5(1) + 1 = 0.5 > 0$. Here, $w = -0.5$ and $bx = 1(1) = 1$. Since $-0.5 < 1$, the elbow should move left.

$$\text{elbow} = -\frac{1}{-0.5} = 2$$

$$w_{\text{new}} = -0.5 - 0.1(1) = -0.6$$

$$b_{\text{new}} = 1 - 0.1 = 0.9$$

$$\text{elbow}_{\text{new}} = -\frac{0.9}{-0.6} = 1.5$$

Since $1.5 < 2$, the elbow **moves left**. The numeric check is consistent with the analytical derivation for this case.

Parameter	Change
(a) Slope w	$w_{\text{new}} = w - \lambda x$: Decreases (becomes more negative)
(b) Bias b	$b_{\text{new}} = b - \lambda$: Decreases (since $\lambda > 0$)
(c) Elbow e	Moves left if $w < bx$ (and right if $w > bx$). Numeric check verifies movement left in a typical case.

Export to Sheets



Prompting Gemini one more time by being explicit with steps, it is able to solve the problem correctly

Do ONLY case (iv): w is negative, x is positive, and the unit is active ($wx+b > 0$). The prediction is 1 too high.

Write the SGD updates explicitly: $w_{\text{new}} = w - lr*x$ and $b_{\text{new}} = b - lr$ (since active and error=+1).

Rewrite the elbow in a sign-safe way for $w < 0$: $\text{elbow} = b/w = b/(-w) = b/|w|$. Do the same for $\text{elbow}_{\text{new}} = b_{\text{new}}/|w_{\text{new}}|$ (as long as w_{new} stays negative).

Now compare $\text{elbow}_{\text{new}}$ to elbow using that form (b decreases, $|w|$ increases) and conclude left/right.

Mention the only caveats:

if the step is huge, w_{new} could cross 0 (elbow formula changes),

or the point could become inactive after the update ($wx+b$ might drop below 0).

Finish with a simple numeric check (e.g., $w=-0.5$, $b=1$, $x=1$, $lr=0.1$) to confirm the direction.

Do NOT derive weird conditions like " $w < b*x$ " unless you verify they can actually happen given $w < 0$ and $x > 0$.

◆ Here is the rigorous analysis for case (iv) following your exact safety and derivation checklist.

5. (b) (iv) ReLU Elbow Update Analysis

Case: $w < 0, x > 0$, and the unit is **active** ($\phi(x) = wx + b > 0$). **Error:** $\delta = \phi(x) - y = +1$ (Prediction is 1 unit too high).

1. Gradients and SGD Updates

Since the unit is active and $\delta = +1$, the gradients are $\frac{\partial \ell}{\partial w} = x$ and $\frac{\partial \ell}{\partial b} = 1$.

$$\begin{aligned}\mathbf{w}_{\text{new}} &= w - \lambda \frac{\partial \ell}{\partial w} = w - \lambda x \\ \mathbf{b}_{\text{new}} &= b - \lambda \frac{\partial \ell}{\partial b} = b - \lambda\end{aligned}$$

2. Analysis of Parameter Changes

1. **Slope w :** Since $w < 0$ and $\lambda x > 0$, $w_{\text{new}} = w - (\text{positive})$.

- w_{new} becomes **more negative**.
- The magnitude $|w|$ **increases**.

2. **Bias b :** Since $\lambda > 0$, $b_{\text{new}} = b - (\text{positive})$.

- b_{new} **decreases**.

3. Sign-Safe Elbow Comparison

Since the unit is active and $w < 0$, the initial bias b must be positive ($b > -wx > 0$). We assume λ is small enough that $b_{\text{new}} > 0$ and $w_{\text{new}} < 0$.

Elbow	Sign-Safe Form (Since $w < 0$ and $w_{\text{new}} < 0$)	Change
elbow	$\frac{b}{ w }$	-
elbow _{new}	$\frac{b_{\text{new}}}{ w_{\text{new}} }$	-

We compare the new elbow to the old one:

elbow_{new} vs elbow

$$\frac{b_{\text{new}}}{|w_{\text{new}}|} \quad \text{vs} \quad \frac{b}{|w|}$$

From the analysis above:

- Numerator $b_{\text{new}} < b$ (**decreases**).
- Denominator $|w_{\text{new}}| > |w|$ (**increases**).

When the numerator of a positive fraction decreases and the denominator increases, the resulting fraction **must decrease**.

Therefore:

This is the correct solution

$$\frac{b_{\text{new}}}{|w_{\text{new}}|} < \frac{b}{|w|} \implies \text{elbow}_{\text{new}} < \text{elbow}$$

Conclusion: The elbow **moves left**.

4. Caveats

1. **Sign flip of w :** If the learning rate λ is huge, w_{new} could cross zero and become positive ($w_{\text{new}} > 0$). If this happens, the standard elbow formula $e = -b/w$ becomes negative, meaning the elbow moves from positive to negative x territory, radically changing the function's behavior.
 2. **Deactivation:** The update could cause the unit to become inactive at the training point x : $w_{\text{new}}x + b_{\text{new}} \leq 0$. If this occurs, the loss at this point will stop contributing to the update in the next iteration.

5. Numeric Check

Check Active Condition: $\phi(x) = -0.5(1) + 1 = 0.5 > 0$ (Initial active) **Check New Active Condition:** $\phi_{\text{new}}(x) = -0.6(1) + 0.9 = 0.3 > 0$ (Stays active)

Elbow Calculation:

By having it test its own proof with numerical examples, Gemini is able to refactor its approach and correctly guide itself to the correct answer

$$\text{elbow} = \frac{1}{|-0.5|} = 2.0$$

$$\text{elbow}_{\text{new}} = \frac{0.9}{|-0.6|} = 1.5$$

Since $1.5 < 2.0$, the elbow **moves left**. This confirms the analytical result.

Results for Case (iv)

Parameter	Change
(a) Slope w	Decreases (becomes more negative; $ w $ increases)
(b) Bias b	Decreases (shifts line downward)
(c) Elbow e	Moves Left ($\text{elbow}_{\text{new}} < \text{elbow}$) because the bias decreases while the magnitude of the slope increases, resulting in a smaller quotient $b_{\text{new}}/ w_{\text{new}} $.

