

Vector calculus derivatives

Vector Calculus Review

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

n

and $A \in \mathbb{R}$

$n \times n$

. For the following parts, before taking any derivatives, identify what the derivative looks like (is it a scalar, vector, or matrix?) and how we calculate each term in the derivative.

Then carefully solve for an arbitrary entry of the derivative, then stack/arrange all of them to get the final result. Note that the convention we will use going forward is that vector derivatives of a scalar (with respect to a column vector) are expressed as a row vector, i.e. $\frac{\partial f}{\partial x} = [\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}]^T$

$\frac{\partial f}{\partial x_1}$

$,$

$\frac{\partial f}{\partial x_2}$

$, \dots,$

$\frac{\partial f}{\partial x_n}$

] since a row acting on a column gives a scalar. You may have seen alternative conventions before, but the important thing is that you need to understand the types of objects and how they map to the shapes of the multidimensional arrays we use to represent those types.

Homework 0, © Faculty teaching EECS 182, Fall 2025. All Rights Reserved. This may not be publicly shared without explicit permission. 1

Homework 0 @ 2025-08-29 14:33:01-07:00

(a) Show $\frac{\partial}{\partial x}$

$\frac{\partial}{\partial x}$

(x)

$T c) = c$

T

(b) Show $\frac{\partial}{\partial x}$

$\frac{\partial}{\partial x}$

$\|x\|^2$

$2 = 2x$

T

(c) Show $\frac{\partial}{\partial x}$

$\frac{\partial}{\partial x}$

$(Ax) = A$

(d) Show $\frac{\partial}{\partial x}$

$\frac{\partial}{\partial x}$

(x)

$T Ax) = x$

T

$(A + AT$

)

(e) Under what condition is the previous derivative equal to $2x$

$T A?$

Let's go through each part of this **vector calculus review** step by step, identifying the shape (scalar/vector/matrix), calculating the derivative component-wise, and then assembling the final result as a row vector (as per the given convention).

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

Type Analysis:

- $x \in \mathbb{R}^n, c \in \mathbb{R}^n$
- $x^T c$ is a scalar.
- Derivative of scalar w.r.t. column vector \rightarrow row vector $\in \mathbb{R}^{1 \times n}$

Seems able to one shot the problem

Component-wise:

Let $f(x) = \sum_{i=1}^n x_i c_i$, then:

$$\frac{\partial f}{\partial x_i} = c_i \Rightarrow \frac{\partial f}{\partial x} = [c_1, c_2, \dots, c_n] = c^T$$

Final answer:

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

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

Type Analysis:

- $\|x\|_2^2 = x^T x$, a scalar.
- Derivative w.r.t. column vector \rightarrow row vector $\in \mathbb{R}^{1 \times n}$

Component-wise:

Let $f(x) = \sum_{i=1}^n x_i^2$, then:

$$\frac{\partial f}{\partial x_i} = 2x_i \Rightarrow \frac{\partial f}{\partial x} = [2x_1, 2x_2, \dots, 2x_n] = 2x^T$$

Final answer:

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

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

Type Analysis:

- $A \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^n$
- Ax is a vector $\in \mathbb{R}^n$
- Derivative of vector w.r.t. vector \rightarrow matrix $\in \mathbb{R}^{n \times n}$

Component-wise:

Let $y = Ax$, so $y_i = \sum_j A_{ij}x_j$, then:

$$\frac{\partial y_i}{\partial x_j} = A_{ij} \Rightarrow \frac{\partial(Ax)}{\partial x} = A$$

Final answer:

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

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

Type Analysis:

- $x^T Ax$ is a scalar.
- Derivative w.r.t. column vector \rightarrow row vector $\in \mathbb{R}^{1 \times n}$

Approach:

Let $f(x) = x^T Ax$. Then using the **product rule** (or chain rule):

$$\frac{\partial}{\partial x}(x^T Ax) = \left(\frac{\partial}{\partial x} x^T Ax \right)^T = \left(\frac{\partial}{\partial x} (x^T Ax) \right)^T = (A + A^T)x \Rightarrow \left[\frac{\partial}{\partial x}(x^T Ax) \right] = x^T(A + A^T)$$

Alternatively, use index notation:

$$x^T Ax = \sum_{i,j} x_i A_{ij} x_j \Rightarrow \frac{\partial}{\partial x_k} = \sum_j A_{kj} x_j + \sum_i x_i A_{ik} = [Ax]_k + [A^T x]_k \Rightarrow \frac{\partial}{\partial x} = (A + A^T)x \Rightarrow \text{transpose} =$$

✓ Final answer:

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

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

From (d), we have:

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

We want:

$$x^T(A + A^T) = 2x^T A \Rightarrow x^T(A + A^T) = x^T A + x^T A^T = x^T A + (Ax)^T$$

For this to equal $2x^T A$, we must have:

$$x^T A^T = x^T A \Rightarrow x^T A^T = x^T A \Rightarrow (A^T - A)x = 0 \quad \forall x \Rightarrow A^T = A$$

✓ Final answer:

$$\boxed{\text{When } A \text{ is symmetric (i.e., } A^T = A\text{), then } \frac{\partial}{\partial x}(x^T Ax) = 2x^T A}$$

Let me know if you want the matrix calculus rules summarized in a cheat sheet.

Great! These are amazing and one shot. Now solve the SVD problem: 3. Least Squares and the Min-norm problem from the Perspective of SVD

Consider the equation $Xw = y$, where $X \in \mathbb{R}$

$m \times n$

is a non-square data matrix, w is a weight vector, and y is vector of labels corresponding to the datapoints in each row of X .

Let's say that $X = U\Sigma V^T$

is the (full) SVD of X . Here, U and V are orthonormal square matrices, and Σ

is an $m \times n$ matrix with non-zero singular values (σ_i) on the "diagonal".

For this problem, we define Σ^\dagger

an $n \times m$ matrix with the reciprocals of the singular values

$(\frac{1}{\sigma_i})$

along the

"diagonal".

(a) First, consider the case where $m > n$, i.e. our data matrix

X has more rows than columns (tall matrix) and the system is overdetermined. How do we find the weights w that minimizes the error between Xw and y ? In other words, we want to solve $\min_w \|Xw - y\|_2^2$

(b) Plug in the SVD $X = U\Sigma V^T$ and simplify. Be careful with dimensions!

(c) You'll notice that the least-squares solution is in the form $w^* = A^{-1}y$. What happens if we left-multiply X by our matrix A? This is why the matrix A of the least-squares solution is called the left-inverse.

(d) Now, let's consider the case where $m < n$, i.e. the data matrix X has more columns than rows and the system is underdetermined. There exist infinitely many solutions for w, but we seek the minimum-norm solution, i.e. we want to solve $\min_w \|w\|_2^2$ s.t. $Xw = y$. What is the minimum norm solution?

(e) Plug in the SVD $X = U\Sigma V^T$ and simplify. Be careful with dimensions!

(f) You'll notice that the min-norm solution is in the form $w^* = B^{-1}y$. What happens if we right-multiply X by our matrix B? This is why the matrix B of the min-norm solution is called the right-inverse.

Thanks! Let's solve the **SVD Least Squares and Min-norm** problem step by step using the tools of linear algebra and matrix calculus. We'll work through parts (a)–(f) with clear reasoning and dimension checks.

Setup:

We are given:

- $X \in \mathbb{R}^{m \times n}$
- Full SVD: $X = U\Sigma V^T$
 - $U \in \mathbb{R}^{m \times m}$, orthonormal
 - $V \in \mathbb{R}^{n \times n}$, orthonormal
 - $\Sigma \in \mathbb{R}^{m \times n}$, diagonal "matrix" with singular values $\sigma_1, \dots, \sigma_r$ (where $r = \text{rank}(X)$)

Let:

- $\Sigma^\dagger \in \mathbb{R}^{n \times m}$ be the **Moore-Penrose pseudo-inverse** of Σ

(a) Tall matrix (overdetermined): Solve

$$\min_w \|Xw - y\|_2^2$$

This is the classic **least squares** problem. When $X \in \mathbb{R}^{m \times n}$ with $m > n$, and X has full column rank:

The closed-form solution is:

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

This solution is right

This minimizes the squared error between the predictions Xw and the true labels $y \in \mathbb{R}^m$.

█ (b) Plug in the SVD: $X = U\Sigma V^T$

Substitute into the least squares formula:

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

Since U is orthonormal: $U^T U = I_m$

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

Use identity: $(V A V^T)^{-1} = V(A^{-1})V^T$ if V is orthonormal

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

But $(\Sigma^T \Sigma)^{-1} \Sigma^T = \Sigma^\dagger$ (by definition of pseudo-inverse)

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

Generally correct, added additional y term

This is the **compact and stable SVD-based expression** for the least-squares solution.

█ (c) Define $A = V \Sigma^\dagger U^T$. What is AX ?

We are told $w^* = Ay$. Compute:

$$AX = V \Sigma^\dagger U^T (U \Sigma V^T) = V \Sigma^\dagger U^T U \Sigma V^T = V(\Sigma^\dagger \Sigma)V^T$$

Now, since $\Sigma^\dagger \Sigma = I_n$ (identity on the **column space** of X), we get:

$$AX = V I V^T = VV^T$$

Note:

- If X is **full rank** and $m > n$, then $VV^T = I_n$
- More generally, AX is the **orthogonal projector** onto the column space of X

That's why A is called the **left-inverse** of X :

$AX = I_n \quad (\text{on the range of } X)$

█ (d) Now the underdetermined case: $m < n$

We're solving:

$$\min_w \|w\|_2^2 \quad \text{s.t.} \quad Xw = y$$

There are infinitely many solutions (since there are more unknowns than equations).

Among all solutions w that satisfy $Xw = y$, we want the one with **minimum norm**.

The solution is:

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

This minimizes $\|w\|_2$ while still solving the constraint.

█ (e) Plug in SVD: $X = U\Sigma V^T$

Start with:

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

Since $U^T U = I$:

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

Now notice:

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

So:

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

Simplify X, but it gave a y term

💡 Same expression as in the least squares case! The SVD gives us a unified framework to express both solutions.

■ (f) Let $B = V\Sigma^\dagger U^T$. What is XB ?

Recall $X = U\Sigma V^T$, so:

$$XB = (U\Sigma V^T)(V\Sigma^\dagger U^T) = U\Sigma V^T V\Sigma^\dagger U^T = U(\Sigma\Sigma^\dagger)U^T$$

Here, $\Sigma\Sigma^\dagger \in \mathbb{R}^{m \times m}$, and acts as a **projector** onto the row space of X .

So:

$$XB = UU^T$$

This is the **projection matrix onto the row space of X** .

✓ That's why B is the **right-inverse**:

$$XB = I_m \quad (\text{on the row space of } X)$$

✓ Summary Table

Case	Objective	Solution	Expression via SVD	Result of Applying Inverse
Overdetermined ($m > n$)	Minimize $\ Xw - y\ $	$w^* = (X^T X)^{-1} X^T y$	$w^* = V\Sigma^\dagger U^T y$	$AX = VV^T$ (left-inverse)
Underdetermined ($m < n$)	Minimize $\ w\ $ s.t. $Xw = y$	$w^* = X^T (X X^T)^{-1} y$	$w^* = V\Sigma^\dagger U^T y$	$XB = UU^T$ (right-inverse)

Let me know if you'd like a diagram or cheat sheet for this too!

it says simplify though, why are you writing some of the solutions in terms of y?

Great observation — you're absolutely right to call that out. Let's clarify the **difference between expressing solutions in terms of y vs. simplifying purely symbolically**.

In parts (b) and (e), the problem says "**Plug in the SVD and simplify**", and the goal is to **rewrite the solution in terms of U, Σ, V — not in terms of y** . Writing the final answer as $w^* = V\Sigma^\dagger U^T y$ is already a *simplification* of expressions like $(X^T X)^{-1} X^T y$ using the SVD.

However, your point is: **can we go further and simplify without mentioning y at all?** These let's go back and re-express (b) and (e) purely in terms of matrices — no y , just simplified matrix forms.

➡ Revised (b): Overdetermined Case — Simplify SVD

Original:

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

Plug in SVD:

$$\text{Let } X = U \Sigma V^T$$

So:

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

Also:

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

So the whole expression becomes:

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

Now notice:

$$(\Sigma^T \Sigma)^{-1} \Sigma^T = \Sigma^\dagger \quad (\text{by def})$$

Thus:

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

Upon reprompting it is now right

Fully simplified: the matrix-only form of the least squares solution operator is:

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

No `y` mentioned — just the simplification of the inverse-expression using the SVD.

⌚ Revised (e): Underdetermined Case — Simplify SVD

Original:

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

Again, plug in $X = U \Sigma V^T$

Then:

- $X X^T = U \Sigma V^T V \Sigma^T U^T = U \Sigma \Sigma^T U^T$
- So:

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

Also:

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

Now:

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

Recognizing again that:

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

We get:

These

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

So the simplified matrix expression for the **minimum-norm operator** is again:

$$B = V\Sigma^{\dagger}U^T$$

Same operator, different motivation.

Why This Is Useful:

Both the overdetermined and underdetermined cases result in the **same simplified matrix expression**:

$$V\Sigma^{\dagger}U^T$$

This is the **Moore–Penrose pseudoinverse** of X :

$$X^{\dagger} = V\Sigma^{\dagger}U^T$$

So:

- In the least squares case: $w^* = X^{\dagger}y$
- In the min-norm case: $w^* = X^{\dagger}y$

The difference lies in:

- Whether X is tall (left-invertible)
- Or wide (right-invertible)

Final Answers (Purely Matrix Simplified):

(b)

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

(e)

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

Let me know if you want visualizations or numerical examples to test this on real matrices.

4. The 5 Interpretations of Ridge Regression

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

argmin_w

$\|y - Xw\|_2^2$

$+ \lambda \|w\|_2^2$

$\frac{1}{2}$

$\frac{1}{2}$

, (1)

where $X \in \mathbb{R}^{n \times d}$

n

is a data matrix, and $y \in \mathbb{R}^n$

n

is the target vector of measurement values. What's new compared to the simple OLS problem is the addition of the $\lambda \|w\|_2^2$

$\frac{1}{2}$

term, which can be interpreted as a "penalty" on the weights being too big.

Use vector calculus to expand the objective and solve this

optimization problem for w.

(b) Perspective 2: "Hack" of shifting the Singular Values. In the previous part, you should have found the optimal w is given by

$$w = (X^T X + \lambda I)$$

$$-1X^T y$$

(If you didn't get this, you should check your work for the previous part).

Homework 0, © Faculty teaching EECS 182, Fall 2025. All Rights Reserved. This may not be publicly shared without explicit permission. 2

Homework 0 @ 2025-08-29 14:33:01-07:00

Let $X = U\Sigma V$

T be the (full) SVD of the X. Recall that U and V are square orthonormal (normpreserving) matrices, and Σ is a $n \times d$ matrix with singular values σ_i along the "diagonal". Plug this into the Ridge Regression solution and simplify. What happens to the singular values of

$$(X^T X + \lambda I)$$

$-1X^T$ when $\sigma_i << \lambda$? What about when $\sigma_i >> \lambda$?

(c) Perspective 3: Maximum A Posteriori (MAP) estimation. Ridge Regression can be viewed as finding

the MAP estimate when we apply a prior on the (now viewed as random parameters) W. In particular, we can think of the prior for W as being $N(0, I)$ and view the random Y as being generated using

$$Y = x$$

$$TW +$$

$\sqrt{\cdot}$

λN where the noise N is distributed iid (across training samples) as $N(0, 1)$. At the vector level, we have $Y = XW +$

$\sqrt{\cdot}$

λN . Note that the X matrix whose rows are the n different training points are not random.

Show that (1) is the MAP estimate for W given an observation $Y = y$.

(d) Perspective 4: Fake Data. Another way to interpret "ridge regression" is as the ordinary least squares for an augmented data set — i.e. adding a bunch of fake data points to our data. Consider the following augmented measurement vector \hat{y} and data matrix \hat{X} :

$$\hat{y} =$$

"

y

0d

#

$$\hat{X} =$$

"

X

$\sqrt{\cdot}$

$\lambda I d$

#

'

where 0d is the zero vector in R

d

and $I d \in \mathbb{R}$

$d \times d$

is the identity matrix. Show that the classical OLS optimization problem $\operatorname{argmin}_w \| \hat{y} - \hat{X}w \|_2^2$

2

has the same minimizer as (1).

(e) Perspective 5: Fake Features. For this last interpretation, let's instead construct an augmented design matrix in the following way:

$$X^* = [X$$

\sqrt

$$\lambda I_n]$$

i.e. we stack X with $\sqrt{\lambda I_n}$ horizontally. Now our problem is underdetermined: the new dimension

$d + n$ is larger than the number of points n . Therefore, there are infinitely many values $\eta \in \mathbb{R}^{d+n}$

for

which $X^* \eta = y$. We are interested in the min-norm solution, i.e. the solution to

$\operatorname{argmin}_{\eta}$

$$\|\eta\|_2^2$$

$$2$$

$$2$$

s.t. $X^* \eta = y$. (2)

Show that this is yet another form of ridge regression and that the first d coordinates of η

*

form

the minimizer of (1).

(f) We know that the Moore-Penrose pseudo-inverse for an underdetermined system (wide matrix) is given by $A^\dagger = A^T A^{-1}$

$(A^T A)^{-1}$

)

-1

, which corresponds to the min-norm solution for $A\eta = z$.

That is, the

optimization problem

$\operatorname{argmin}_{\eta} \|A\eta - z\|_2^2$

2

s.t. $A\eta = z$

is solved by $\eta = A^T z$. Let \hat{w} be the minimizer of (1).

Use the pseudo-inverse to show that solving to the optimization problem in (2) yields

$$\hat{w} = X^T$$

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

$$-1$$

Then, show that this is equivalent to the standard formula for Ridge Regression

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

Hint: It may be helpful to review Kernel Ridge Form.

Homework 0, © Faculty teaching EECS 182, Fall 2025. All Rights Reserved. This may not be publicly shared without explicit permission. 3

Homework 0 @ 2025-08-29 14:33:01-07:00

(g) We know that the solution to ridge regression (1) is given by $\hat{w} = (X^T X + \lambda I)^{-1} X^T y$. What

happens when $\lambda \rightarrow \infty$? It is for this reason that sometimes ridge regularization is referred to as "shrinkage."

(h) What happens to the solution of ridge regression when you take the limit $\lambda \rightarrow 0$? Consider both

the cases when X is wide (underdetermined system) and X is tall (overdetermined system).

Awesome — this is the **Ridge Regression: 5 Perspectives** problem from EECS 182 (Fall 2025). Let's walk through each part (a)–(h) with step-by-step reasoning.

(a) Perspective 1: Optimization Problem

We are solving:

$$\min_w \|y - Xw\|^2 + \lambda \|w\|^2$$

Let's write the objective function explicitly:

$$L(w) = (y - Xw)^T(y - Xw) + \lambda w^T w = y^T y - 2y^T Xw + w^T X^T Xw + \lambda w^T w$$

Now take derivative w.r.t. $w \in \mathbb{R}^d$ (as a **row vector**):

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

Set gradient to 0:

$$2w^T(X^T X + \lambda I) = 2y^T X \Rightarrow w^T = y^T X(X^T X + \lambda I)^{-1} \Rightarrow \boxed{w = (X^T X + \lambda I)^{-1} X^T y}$$

(b) Perspective 2: Shifting Singular Values

Given:

- $X = U\Sigma V^T$
- Ridge solution:

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

Use the identity:

$$X^T X = V\Sigma^T \Sigma V^T \Rightarrow X^T X + \lambda I = V(\Sigma^T \Sigma + \lambda I)V^T$$

So:

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

Effect of λ on singular values:

Let singular values be σ_i , then:

$$\frac{\sigma_i}{\sigma_i^2 + \lambda}$$

- If $\sigma_i \ll \lambda$: denominator is dominated by $\lambda \Rightarrow$ expression \approx small \Rightarrow **singular directions with small σ get suppressed**
 - If $\sigma_i \gg \lambda$: denominator $\approx \sigma_i^2 \Rightarrow$ expression $\approx \frac{1}{\sigma_i} \Rightarrow$ close to OLS
 - Ridge shrinks contributions from small singular directions, improving numerical stability.
-

(c) Perspective 3: MAP Estimation

These are mostly right

Given:

- Prior: $W \sim \mathcal{N}(0, I)$
- Observation model: $Y = XW + \sqrt{\lambda}N$, where $N \sim \mathcal{N}(0, I)$

MAP estimate maximizes:

$$\log P(W | Y) \propto \log P(Y | W) + \log P(W)$$

Log-likelihood:

$$\log P(Y | W) \propto -\frac{1}{2\lambda} \|Y - XW\|^2 \quad (\text{Gaussian noise})$$

Prior:

$$\log P(W) \propto -\frac{1}{2} \|W\|^2$$

So the MAP objective is:

$$\min_W \left(\frac{1}{2\lambda} \|Y - XW\|^2 + \frac{1}{2} \|W\|^2 \right) \Rightarrow \min_W \|Y - XW\|^2 + \lambda \|W\|^2$$

Matches ridge regression.

MAP estimate = Ridge solution

(d) Perspective 4: Fake Data

Augment data:

$$\hat{y} = \begin{bmatrix} y \\ 0_d \end{bmatrix}, \quad \hat{X} = \begin{bmatrix} X \\ \sqrt{\lambda} I_d \end{bmatrix}$$

Now solve:

$$\min_w \|\hat{y} - \hat{X}w\|^2 = \|y - Xw\|^2 + \|\sqrt{\lambda}w\|^2 = \|y - Xw\|^2 + \lambda \|w\|^2$$

So this is exactly ridge regression.

OLS on $(\hat{X}, \hat{y}) \Rightarrow$ ridge regression solution

(e) Perspective 5: Fake Features (Min-norm form)

Augment horizontally:

$$\check{X} = [X \quad \sqrt{\lambda} I_n] \in \mathbb{R}^{n \times (d+n)}$$

Define variable $\eta \in \mathbb{R}^{d+n}$, and solve:

$$\min_{\eta} \|\eta\|^2 \quad \text{s.t. } \check{X}\eta = y$$

Write $\eta = \begin{bmatrix} w \\ z \end{bmatrix}$, then:

$$\check{X}\eta = Xw + \sqrt{\lambda}z = y \Rightarrow z = \frac{1}{\sqrt{\lambda}}(y - Xw)$$

So the objective becomes:

$$\|\eta\|^2 = \|w\|^2 + \|z\|^2 = \|w\|^2 + \frac{1}{\lambda} \|y - Xw\|^2$$

Therefore:

$$\min_{\eta} \|\eta\|^2 \Leftrightarrow \min_w \|y - Xw\|^2 + \lambda \|w\|^2$$

The first d entries of the min-norm solution η^* correspond to ridge regression solution \hat{w}

(f) Use Pseudo-inverse to show kernel form of ridge:

From part (e), you solve:

$$\min_{\eta} \|\eta\|^2 \quad \text{s.t. } \check{X}\eta = y \Rightarrow \eta = \check{X}^\dagger y = \check{X}^T (\check{X}\check{X}^T)^{-1} y$$

$$\text{Recall } \check{X} = [X \quad \sqrt{\lambda}I_n]$$

Then:

$$\check{X}\check{X}^T = XX^T + \lambda I$$

So:

$$\eta = \check{X}^T (XX^T + \lambda I)^{-1} y$$

First d components of η are:

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

This is the **dual/kernel form** of ridge regression.

Now prove equivalence to standard form:

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

Use matrix identity:

$$(X^T X + \lambda I)^{-1} X^T = X^T (XX^T + \lambda I)^{-1} \Rightarrow (X^T X + \lambda I)^{-1} X^T y = X^T (XX^T + \lambda I)^{-1} y$$

Proven!

(g) Behavior as $\lambda \rightarrow \infty$

Ridge regression solution:

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

As $\lambda \rightarrow \infty$:

- λI dominates \rightarrow inverse tends to 0
- So $\hat{w} \rightarrow 0$

This is called **shrinkage**: all weights are pushed toward zero.

$$\hat{w} \rightarrow 0 \text{ as } \lambda \rightarrow \infty$$

(h) Behavior as $\lambda \rightarrow 0$

Case 1: Tall matrix ($n > d$)

- $X^T X$ is full-rank and invertible
- So:

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

Converges to OLS solution

Case 2: Wide matrix ($n < d$)

- $X^T X$ is not invertible
- So OLS is ill-posed (infinitely many solutions)
- But ridge still yields a unique solution

As $\lambda \rightarrow 0$:

- \hat{w} tends toward a solution of $Xw = y$
 - But among infinitely many solutions, it converges to the **min-norm solution**
- So:
- | |
|--|
| $\hat{w} \rightarrow \text{OLS solution if unique (tall)}$ |
| $\hat{w} \rightarrow \text{minimum-norm solution if underdetermined (wide)}$ |

Let me know if you want a PDF-style writeup or a diagram showing how all 5 perspectives are connected.

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 how SGD works and how it iteratively adjusts the learned function.

We want to model a 1D function $y = f(x)$ using a 1-hidden layer network with ReLU activations and no biases in the linear output layer. Mathematically, our network is

$$\hat{f}(x) = W(2)\phi$$

$$W(1)x + b$$

where $x, y \in \mathbb{R}$, $b \in \mathbb{R}$

d

, $W(1) \in \mathbb{R}$

$d \times 1$

, and $W(2) \in \mathbb{R}$

$1 \times d$

. We define our loss function to be the squared error,

ℓ

$x, y, W(1)$

, $b, W(2)$

=

1

2

$$\hat{f}(x) - y$$

2

2

.

For the purposes of this problem, we define the gradient

of a ReLU at 0 to be 0.

(a) Let's start by examining the behavior of a single ReLU with a linear function of x as the input,

$$\phi(x) = \begin{cases} wx + b, & wx + b > 0 \\ 0, & \text{else} \end{cases}$$

Notice that the slope of $\phi(x)$ is w in the non-zero domain.

We define a loss function $\ell(x, y, \phi) = 1$

$$\begin{aligned} & 2 \\ & \| \phi(x) - y \| \\ & 2 \\ & 2 \end{aligned}$$

. Find the following:

(i) The location of the 'elbow' e of the function, where it transitions from 0 to something else.

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

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

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

(b) Now suppose we have some training point (x, y) such that $\phi(x) - y = 1$. In other words, the prediction $\phi(x)$ is 1 unit above the target y — we are too high and are trying to pull the function downward.

Describe what happens to the slope and elbow of $\phi(x)$ when we perform gradient descent in the following cases:

(i) $\phi(x) = 0$.

(ii) $w > 0, x > 0$, and $\phi(x) > 0$. It is fine to check the behavior of the elbow numerically in this case.

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

(iv) $w < 0, x > 0$, and $\phi(x) > 0$. It is fine to check the behavior of the elbow numerically in this case.

Homework 0, © Faculty teaching EECS 182, Fall 2025. All Rights Reserved. This may not be publicly shared without explicit permission. 4

Homework 0 @ 2025-08-29 14:33:01-07:00

Additionally, draw and label $\phi(x)$, the elbow, and the qualitative changes to the slope and elbow after a gradient update to w and b . You should label the elbow location and a candidate (x, y) pair. Remember that the update for some parameter vector p and loss ℓ under SGD is

$$p' = p - \lambda \nabla p(\ell), \lambda > 0.$$

(c) Now we return to the full network function $\hat{f}(x)$. Derive the location e_i of the elbow of the i 'th elementwise ReLU activation.

(d) Derive the new elbow location e'_i

i

of the i 'th elementwise ReLU activation after one stochastic gradient update with learning rate λ .

Let's work through the **ReLU Elbow Update under SGD** problem in EECS 182 step by step. We'll handle all four subparts (a)–(d) with clear derivations and explanations — including formulas, intuitions, and how the elbow changes under SGD.

(a) Single ReLU Unit

We are given the activation function:

$$\varphi(x) = \begin{cases} wx + b & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

and the loss:

$$\ell(x, y, \varphi) = \frac{1}{2}(\varphi(x) - y)^2$$

(i) Location of the "elbow" e

The **elbow** is where $\varphi(x)$ transitions from 0 to non-zero — i.e., the argument to ReLU becomes 0:

$$wx + b = 0 \Rightarrow x = -\frac{b}{w}$$

 Answer:

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

(ii) Derivative of loss w.r.t. output $\varphi(x)$

We define:

$$\ell = \frac{1}{2}(\varphi(x) - y)^2 \Rightarrow \frac{d\ell}{d\varphi(x)} = \varphi(x) - y$$

 Answer:

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

(iii) Partial derivative of loss w.r.t. w

Use chain rule:

$$\frac{\partial \ell}{\partial w} = \frac{d\ell}{d\varphi(x)} \cdot \frac{\partial \varphi(x)}{\partial w}$$

Case 1: If $wx + b \leq 0 \Rightarrow \varphi(x) = 0$, and ReLU derivative is defined to be 0 at 0, so:

$$\frac{\partial \varphi(x)}{\partial w} = 0 \Rightarrow \frac{\partial \ell}{\partial w} = 0$$

Case 2: If $wx + b > 0 \Rightarrow \varphi(x) = wx + b$, so:

$$\frac{\partial \varphi(x)}{\partial w} = x \Rightarrow \frac{\partial \ell}{\partial w} = (\varphi(x) - y) \cdot x$$

 Answer:

$$\frac{\partial \ell}{\partial w} = \begin{cases} 0 & \text{if } wx + b \leq 0 \\ (\varphi(x) - y) \cdot x & \text{if } wx + b > 0 \end{cases}$$

(iv) Partial derivative of loss w.r.t. b

Same reasoning as above:

$$\frac{\partial \ell}{\partial b} = \frac{d\ell}{d\varphi(x)} \cdot \frac{\partial \varphi(x)}{\partial b}$$

- If $wx + b \leq 0$, then derivative is 0.
- If $wx + b > 0$, then $\frac{\partial \varphi(x)}{\partial b} = 1$

Answer:

$$\frac{\partial \ell}{\partial b} = \begin{cases} 0 & \text{if } wx + b \leq 0 \\ (\varphi(x) - y) & \text{if } wx + b > 0 \end{cases}$$

■ (b) Behavior of Gradient Descent on Elbow & Slope

Given: $\varphi(x) - y = 1$, so we are **too high** and want to decrease $\varphi(x)$

Let's study what happens to w and b under SGD:

$$w' = w - \lambda \cdot \frac{\partial \ell}{\partial w} \quad b' = b - \lambda \cdot \frac{\partial \ell}{\partial b}$$

Also recall:

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

✓ (i) Case: $\varphi(x) = 0$

Means: $wx + b \leq 0 \Rightarrow \text{ReLU inactive}$

Then:

- $\frac{\partial \ell}{\partial w} = 0$
- $\frac{\partial \ell}{\partial b} = 0$
- So: $w' = w, b' = b$, and **nothing changes**

Summary:

- **Slope:** unchanged
 - **Elbow:** unchanged
-

✓ (ii) Case: $w > 0, x > 0, \varphi(x) > 0$

Then:

- $\frac{\partial \ell}{\partial w} = (1) \cdot x = x$
- $\frac{\partial \ell}{\partial b} = 1$

Update:

$$w' = w - \lambda x \quad (\downarrow) \quad b' = b - \lambda \quad (\downarrow)$$

Elbow:

$$e = -\frac{b}{w} \Rightarrow e' = -\frac{b - \lambda}{w - \lambda x}$$

Numerical example: say $w = 2, b = 1, x = 1, \lambda = 0.1$

- Old: $e = -\frac{1}{2} = -0.5$
- New: $e' = -\frac{0.9}{1.9} \approx -0.474$

Summary:

- **Slope decreases**
- **Elbow shifts slightly right** (if $x > 0$)

Elbow shifts for (ii) and (iii) are wrong

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

Update:

- $\frac{\partial \ell}{\partial w} = (1) \cdot x < 0 \Rightarrow w' = w - \lambda x \uparrow$
- $b' = b - \lambda$

Elbow:

$$e' = -\frac{b - \lambda}{w + \lambda|x|} \Rightarrow \text{denominator increases, numerator decreases}$$

→ Elbow shifts left

 Summary:

- Slope increases
- Elbow shifts left

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

- $\frac{\partial \ell}{\partial w} = (1) \cdot x > 0 \Rightarrow w' = w - \lambda x \downarrow$
- $b' = b - \lambda$

Now both numerator and denominator decrease (more negative)

$$\rightarrow \text{Elbow } e' = -\frac{b - \lambda}{w - \lambda x}$$

Depending on signs, this might push the elbow right or left, but in general:

 Summary:

- Slope decreases (steeper negative)
- Elbow shifts

 Drawing Guidance

In your sketch:

- Label:
 - Elbow: $x = -b/w$
 - Original and updated lines
 - (x, y) point with error
 - Arrows showing slope and elbow motion

 (c) Elbow location e_i for i-th ReLU in full network

Each hidden unit computes:

$$\varphi_i(x) = \text{ReLU}(w_i^{(1)}x + b_i)$$

Elbow where pre-activation is zero:

$$w_i^{(1)}x + b_i = 0 \Rightarrow x = -\frac{b_i}{w_i^{(1)}}$$

 Answer:

$$e_i = -\frac{b_i}{w_i^{(1)}}$$

 (d) Elbow location after SGD update

Let:

$$w_i^{(1)\prime} = w_i^{(1)} - \lambda \frac{\partial \ell}{\partial w_i^{(1)}} \quad b_i' = b_i - \lambda \frac{\partial \ell}{\partial b_i}$$

Then new elbow:

$$e_i' = -\frac{b_i - \lambda \frac{\partial \ell}{\partial b_i}}{w_i^{(1)} - \lambda \frac{\partial \ell}{\partial w_i^{(1)}}}$$

That's the updated location based on gradient changes to w and b .

Let me know if you want me to render these as plots or provide a Python visualization template!

no (ii) elbow can shift both ways and (iii) elbow shifts right

You're absolutely right to call that out — let's correct the elbow movement logic for parts (ii) and (iii) in the ReLU gradient update question.

Let's now go over each case again carefully, **only focusing on the direction the elbow shifts** under a gradient descent update, with all signs and ratios tracked properly.

 **Reminder:**

The ReLU elbow is at:

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

After a gradient descent step:

$$w' = w - \lambda \frac{\partial \ell}{\partial w}, \quad b' = b - \lambda \frac{\partial \ell}{\partial b} \Rightarrow e' = -\frac{b'}{w'} = -\frac{b - \lambda \frac{\partial \ell}{\partial b}}{w - \lambda \frac{\partial \ell}{\partial w}}$$

Given: $\varphi(x) - y = 1$, so gradient is:

- $\frac{\partial \ell}{\partial w} = x$
 - $\frac{\partial \ell}{\partial b} = 1$
-

 (ii) $w > 0, x > 0, \varphi(x) > 0$

Update:

- $w' = w - \lambda x \rightarrow \text{decreases}$
- $b' = b - \lambda \rightarrow \text{decreases}$

So:

$$e' = -\frac{b - \lambda}{w - \lambda x}$$

Let's analyze how e' compares to $e = -\frac{b}{w}$:

Compute difference:

$$e' - e = -\frac{b - \lambda}{w - \lambda x} + \frac{b}{w}$$

We want to check the sign of this quantity — it depends on the values of b, w, λ, x , so:

Try numerical example:

- $w = 2, b = 1, x = 1, \lambda = 0.1$

Then:

- $e = -\frac{1}{2} = -0.5$
- $w' = 2 - 0.1 = 1.9$
- $b' = 1 - 0.1 = 0.9$
- $e' = -\frac{0.9}{1.9} \approx -0.474$

So:

$$e' > e \Rightarrow \text{Elbow shifts **right**}$$

Corrected answer:

- Slope decreases
- Elbow shifts right

(iii) $w > 0, x < 0, \varphi(x) > 0$

Update:

- $\frac{\partial \ell}{\partial w} = x < 0 \Rightarrow w' = w - \lambda x = w + \lambda|x| \rightarrow \text{increases}$
- $b' = b - \lambda \rightarrow \text{decreases}$

So:

$$e' = -\frac{b - \lambda}{w + \lambda|x|}$$

Try a numerical example:

- $w = 2, b = 1, x = -1, \lambda = 0.1$

Then:

- $e = -\frac{1}{2} = -0.5$
- $w' = 2 + 0.1 = 2.1$
- $b' = 1 - 0.1 = 0.9$
- $e' = -\frac{0.9}{2.1} \approx -0.429$

So:

$$e' > e \Rightarrow \text{Elbow shifts **right**}$$

Corrected answer:

- Slope increases
- Elbow shifts right

Summary (Corrected):

Case	Conditions	Slope Update	Elbow Shift	
(i)	$\varphi(x) = 0$	No change	No change	Elbow shifts are wrong
(ii)	$w > 0, x > 0$	Decreases	Right	
(iii)	$w > 0, x < 0$	Increases	Right	

Case	Conditions	Slope Update	Elbow Shift
(iv)	$w < 0, x > 0$	Decreases further (more negative)	Can shift left or right depending on ratio

Thanks for catching that — especially the elbow **always shifting right** in both (ii) and (iii) under these update dynamics. Let me know if you want to visualize all these cases!

no (ii) your expression was right, but the elbow shift can be one of two ways

You're absolutely right again — let's take a second, more precise pass at (ii) and settle this fully.

⌚ Revisit: (ii) $w > 0, x > 0, \varphi(x) > 0$

We are analyzing:

$$e = -\frac{b}{w}, \quad e' = -\frac{b - \lambda}{w - \lambda x}$$

Our goal: determine when $e' > e$ or $e' < e$ — i.e., which direction the **elbow shifts** after one SGD step with learning rate λ .

🧠 Let's Analyze the Sign of $e' - e$

We compute:

$$e' - e = -\frac{b - \lambda}{w - \lambda x} + \frac{b}{w}$$

Let's define this expression and analyze it:

$$\Delta e := e' - e = \frac{b}{w} - \frac{b - \lambda}{w - \lambda x}$$

Let's write both terms with a common denominator:

$$\Delta e = \frac{b(w - \lambda x) - (b - \lambda)w}{w(w - \lambda x)} = \frac{bw - b\lambda x - bw + \lambda w}{w(w - \lambda x)} = \frac{\lambda(w - bx)}{w(w - \lambda x)}$$

So:

$$\Delta e = e' - e = \frac{\lambda(w - bx)}{w(w - \lambda x)}$$

⌚ Now analyze the sign of Δe :

- $\lambda > 0$
- $w > 0, x > 0$
- So the denominator $w(w - \lambda x) > 0$

Thus, the **sign of Δe** is determined by the numerator:

$$\text{Sign}(e' - e) = \text{sign}(w - bx)$$

⌚ So:

- If $w > bx \rightarrow \Delta e > 0 \rightarrow e' > e \rightarrow$ **elbow shifts right**
- If $w < bx \rightarrow \Delta e < 0 \rightarrow e' < e \rightarrow$ **elbow shifts left**

Final Corrected Answer for (ii):

Under the update:

$$w' = w - \lambda x, \quad b' = b - \lambda \quad \Rightarrow \quad e' = -\frac{b - \lambda}{w - \lambda x}$$

Then:

$$e' \geq e \iff w \geq bx$$

So the elbow can shift left or right depending on the relationship between w and bx .

Summary (Updated Table):

Case	Conditions	Slope Update	Elbow Shift
(i)	$\varphi(x) = 0$	No change	No change
(ii)	$w > 0, x > 0, \varphi(x) > 0$	Decreases	Left or Right depending on $w \geq bx$
(iii)	$w > 0, x < 0$	Increases	Right
(iv)	$w < 0, x > 0$	Decreases	Mixed — depends on update ratio

Thanks for the correction — (ii) is a subtle but important case! Let me know if you'd like plots to visualize this or a symbolic manipulation notebook.

(iv) is also wrong it is supposed to JUST move left

Excellent, you're absolutely correct again — let's do a clean and final pass on Case (iv), double-checking all signs carefully to confirm that the elbow **only moves left** in this situation.

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

We're analyzing the elbow location:

$$e = -\frac{b}{w}, \quad e' = -\frac{b - \lambda}{w - \lambda x}$$

Let's again analyze:

$$\Delta e = e' - e = \frac{\lambda(w - bx)}{w(w - \lambda x)}$$

Same derivation as before, but now apply to signs.

 **Let's Analyze the Signs**

Given:

- $\lambda > 0$
- $x > 0$
- $w < 0 \Rightarrow$ ReLU is "facing left"
- $\varphi(x) = wx + b > 0 \Rightarrow wx + b > 0$

So:

$$b > -wx \Rightarrow bx > -wx^2 \Rightarrow w - bx < 0 \quad (\text{numerator} < 0)$$

Now let's look at the denominator:

$$w < 0, \quad \lambda x > 0 \Rightarrow w - \lambda x < 0 \Rightarrow w(w - \lambda x) > 0 \quad (\text{product of two negatives})$$

So:

- Numerator: < 0
- Denominator: > 0
- $\Rightarrow \Delta e = e' - e < 0 \Rightarrow e' < e$

So the elbow always shifts left under these conditions.

Final Verdict for (iv):

$$w < 0, x > 0, \varphi(x) > 0 \Rightarrow \boxed{\text{Elbow shifts left}}$$

Final Final Table (Everything Corrected):

Case	Conditions	Slope Update	Elbow Shift
(i)	$\varphi(x) = 0$	No change	No change
(ii)	$w > 0, x > 0$	Decreases	Left or Right depending on $w \geqslant bx$
(iii)	$w > 0, x < 0$	Increases	Right
(iv)	$w < 0, x > 0$	Decreases (steeper negative)	Left <input checked="" type="checkbox"/>

Finally corrected, although the reasoning is not that clear.

Thanks for sticking with the details — these sign-and-slope behaviors are core to understanding SGD + ReLU behavior during training. Let me know if you want animated plots or Python simulations of these elbows moving in real time.

could you derive part (d) in more detail please

Absolutely — let's go through part (d) in full detail:

(d) Derive the new elbow location e'_i of the i-th elementwise ReLU activation after one stochastic gradient update with learning rate λ .

Detailed derivation here which is correct, but I had to prompt it

Recap from earlier parts

We are modeling a 1D input $x \in \mathbb{R}$ with a 1-hidden-layer ReLU neural network, no bias at the output layer. The forward pass is:

$$\hat{f}(x) = W^{(2)} \cdot \Phi(W^{(1)}x + b)$$

Where:

- $W^{(1)} \in \mathbb{R}^{d \times 1}$ — input-to-hidden weights
- $b \in \mathbb{R}^d$ — hidden layer biases
- $W^{(2)} \in \mathbb{R}^{1 \times d}$ — hidden-to-output weights
- $\Phi(\cdot)$ — elementwise ReLU
- So the pre-activation of the i -th neuron is:

$$z_i(x) = w_i^{(1)}x + b_i$$

and the elbow is where this **transitions from 0 to non-zero** (i.e., when $z_i(x) = 0$)

Thus:

$$\text{elbow of neuron } i : e_i = -\frac{b_i}{w_i^{(1)}}$$

Now: After one SGD step

We assume the network is trained using **squared loss**:

$$\ell(x, y) = \frac{1}{2}(\hat{f}(x) - y)^2$$

We're doing **SGD updates** for parameters $w_i^{(1)}$ and b_i . After one SGD step with learning rate λ :

$$w'_i = w_i - \lambda \cdot \frac{\partial \ell}{\partial w_i} \quad \text{and} \quad b'_i = b_i - \lambda \cdot \frac{\partial \ell}{\partial b_i}$$

Then, the **new elbow location** becomes:

$$e'_i = -\frac{b'_i}{w'_i} = -\frac{b_i - \lambda \cdot \frac{\partial \ell}{\partial b_i}}{w_i - \lambda \cdot \frac{\partial \ell}{\partial w_i}}$$

Let's now derive $\frac{\partial \ell}{\partial w_i}$ and $\frac{\partial \ell}{\partial b_i}$

We'll use **chain rule** through the network. Let's define:

- $z_i = w_i x + b_i$
- $\phi_i = \text{ReLU}(z_i)$
- $\hat{f}(x) = \sum_{j=1}^d W_j^{(2)} \phi_j$

Suppose $\phi_i > 0$ (i.e., ReLU is **active**, so its derivative is 1)

Then:

$$\frac{\partial \ell}{\partial \phi_i} = \frac{\partial \ell}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial \phi_i} = (\hat{f} - y) \cdot W_i^{(2)}$$

Next:

$$\frac{\partial \phi_i}{\partial z_i} = 1[z_i > 0] = 1 \quad (\text{since active}) \Rightarrow \frac{\partial \ell}{\partial z_i} = (\hat{f} - y) W_i^{(2)}$$

Now apply:

$$\frac{\partial z_i}{\partial w_i} = x \quad \text{and} \quad \frac{\partial z_i}{\partial b_i} = 1$$

So finally:

$$\frac{\partial \ell}{\partial w_i} = (\hat{f} - y) W_i^{(2)} \cdot x \quad , \quad \frac{\partial \ell}{\partial b_i} = (\hat{f} - y) W_i^{(2)}$$

Now plug into the new elbow expression:

$$e'_i = -\frac{b_i - \lambda(\hat{f} - y)W_i^{(2)}}{w_i - \lambda(\hat{f} - y)W_i^{(2)}x}$$

This is the updated location of the elbow for neuron i after one SGD step, under the assumption that the ReLU is **active at x** (i.e. $z_i(x) > 0$).

 **Edge case: if ReLU is inactive, then gradients are 0**

If $z_i(x) \leq 0 \Rightarrow \phi_i = 0$, then by convention:

$$\frac{\partial \phi_i}{\partial z_i} = 0 \Rightarrow \frac{\partial \ell}{\partial w_i} = 0, \quad \frac{\partial \ell}{\partial b_i} = 0 \Rightarrow w'_i, b'_i = w_i, b_i \Rightarrow e'_i = e_i$$

 **Final Answer (Full Derivation):**

$$e'_i = \begin{cases} -\frac{b_i - \lambda(\hat{f}(x) - y)W_i^{(2)}}{w_i^{(1)} - \lambda(\hat{f}(x) - y)W_i^{(2)}x} & \text{if } w_i^{(1)}x + b_i > 0 \\ e_i & \text{if } w_i^{(1)}x + b_i \leq 0 \end{cases}$$

Let me know if you'd like a plot or animation to visualize how this elbow moves depending on the sign of $\hat{f}(x) - y$, or how this behaves for multiple ReLU neurons.