Note that this file was generated using Claude Opus 4.5 with Extended Reasoning. Claude wrote it's solutions to a markdown file with embedded LaTex and then I compiled said file to a pdf with pandoc.

EECS 182 Deep Neural Networks - Homework 0 Solutions

Math Sections (Questions 2, 3, 4, 5)

Question 2: Vector Calculus Review

Setup: Let $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$.

Convention: Vector derivatives of a scalar with respect to a column vector are expressed as a row vector:

$$\frac{\partial f}{\partial \mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n} \right]$$

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

Step 1: Identify the type of derivative - $\mathbf{x}^T\mathbf{c}$ is a scalar (inner product of two vectors) - The derivative of a scalar w.r.t. a column vector is a **row vector** of size $1 \times n$

Step 2: Expand the expression

$$\mathbf{x}^T\mathbf{c} = \sum_{i=1}^{n} x_i c_i = x_1c_1 + x_2c_2 + \cdots + x_nc_n$$

Step 3: Compute an arbitrary entry of the derivative

$$\frac{\partial}{\partial x_j}(\mathbf{x}^T\mathbf{c}) = \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{n} x_i c_i \right) = c_j$$

Since only the j-th term in the sum contains x_j, and that term is $x_j c_j$.

Step 4: Stack all entries

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{c}) = [c_1, c_2, \ldots, c_n] = \mathbf{c}^T$$

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

Comparing 2.a to the staff solutions, Claude was able to solve this problem without error. It's also worth noting that Claude in this question goes more in depth in it's solutions and steps.

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

Step 1: Identify the type of derivative - $\|\mathbf{x}\|_2^2$ is a scalar - The derivative is a **row vector** of size $1 \times n$

Step 2: Expand the expression

$$\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x} = \sum_{i=1}^{n} x_i^2 = x_1^2 + x_2^2 + \cdots + x_n^2$$

Comparing 2.b to the staff solutions, Claude was able to solve this problem without error.

Step 3: Compute an arbitrary entry of the derivative

$$\frac{\partial}{\partial x_j}\|\mathbf{x}\|_2^2 = \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{n} x_i^2\right) = 2x_j$$

Step 4: Stack all entries

$$\frac{\partial}{\partial \mathbf{x}}\|\mathbf{x}\|_2^2 = [2x_1, 2x_2, \ldots, 2x_n] = 2[x_1, x_2, \ldots, x_n] = 2\mathbf{x}^T$$

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

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

Step 1: Identify the type of derivative - $A\mathbf{x}$ is a vector of size $n \times 1$ - The derivative of a vector w.r.t. a vector is a **matrix** - Specifically: $\frac{\partial (A\mathbf{x})}{\partial \mathbf{x}}$ has entry (i, j) equal to $\frac{\partial (A\mathbf{x})_i}{\partial x_j}$ - This gives an $n \times n$ matrix

Step 2: Expand the i-th component of $A\mathbf{x}$

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

Comparing 2.c to the staff solutions, Claude was able to solve this problem without error.

Step 3: Compute an arbitrary entry of the Jacobian

$$\frac{\partial (A\mathbf{x})_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\sum_{k=1}^{n} A_{ik}x_k\right) = A_{ij}$$

Since only the term with $k = j$ contributes.

Step 4: Assemble the matrix The (i, j) entry of the Jacobian is A_{ij}, which means the Jacobian is exactly A.

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

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

Step 1: Identify the type of derivative - $\mathbf{x}^T A\mathbf{x}$ is a scalar (quadratic form) - The derivative is a **row vector** of size $1 \times n$

Step 2: Expand the expression

$$\mathbf{x}^T A\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i A_{ij} x_j$$

Step 3: Compute an arbitrary entry of the derivative

$$\frac{\partial}{\partial x_k}(\mathbf{x}^T A\mathbf{x}) = \frac{\partial}{\partial x_k}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i A_{ij} x_j\right)$$

We need to consider three cases for terms in the double sum: - Terms where $i = k$ and $j \neq k$: $x_k A_{kj} x_j \rightarrow$ derivative is $A_{kj} x_j$ - Terms where $i \neq k$ and $j = k$: $x_i A_{ik} x_k \rightarrow$ derivative is $x_i A_{ik}$ - Terms where $i = k$ and $j = k$: $x_k A_{kk} x_k = A_{kk} x_k^2$ $\rightarrow$ derivative is $2A_{kk} x_k$

Combining all terms:

$$\frac{\partial}{\partial x_k}(\mathbf{x}^T A\mathbf{x}) = \sum_{j \neq k} A_{kj} x_j + \sum_{i \neq k} x_i A_{ik} + 2A_{kk} x_k$$

$$= \sum_{j=1}^{n} A_{kj} x_j + \sum_{i=1}^{n} x_i A_{ik} = (A\mathbf{x})_k + (A^T\mathbf{x})_k$$

Comparing 2.d to the staff solutions, Claude was able to solve this problem without error. Once again Claude provides more detail and explanation than staff solutions.

Step 4: Stack all entries

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

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

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

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

For this to equal $2\mathbf{x}^T A$, we need:

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

Comparing 2.d to the staff solutions, Claude was able to solve this problem without error.

This requires $A + A^T = 2A$, which simplifies to:

$$A^T = A$$

$$\boxed{\text{The condition is that } A \text{ is symmetric: } A = A^T}$$

Question 3: Least Squares and Min-norm from SVD Perspective

Setup: $X\mathbf{w} = \mathbf{y}$ where $X \in \mathbb{R}^{m \times n}$, $X = U\Sigma V^T$ is the full SVD.

$\Sigma^\dagger$ is $n \times m$ with $\frac{1}{\sigma_i}$ on the "diagonal."

Part (a): Overdetermined case ($m > n$) - Finding least squares solution

When $m > n$, we have more equations than unknowns. We want to minimize:

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

Step 1: Expand the objective

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

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

Comparing 3.a to the staff solutions, Claude was able to solve this problem without error. I also and quite pleased with the notation Claude uses, defining terms at the beginning of quesitons and using consistent notation such as bolded vectors.

Step 2: Take the derivative and set to zero

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

Transposing:

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

Step 3: Solve for w

Assuming $X^T X$ is invertible (which happens when X has full column rank):

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

Part (b): Plug in SVD and simplify

Step 1: Substitute $X = U\Sigma V^T$

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

Note: $U^T U = I$ since U is orthonormal.

Step 2: Compute $\Sigma^T\Sigma$ - Σ is $m \times n$ (with $m > n$) - Σ^T is $n \times m$ - $\Sigma^T\Sigma$ is $n \times n$ diagonal with σ_i^2 on the diagonal

Step 3: Compute $(X^T X)^{-1}$

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

where $(\Sigma^T\Sigma)^{-1}$ is diagonal with $\frac{1}{\sigma_i^2}$ entries.

Comparing 3.b to the staff solutions, Claude was able to solve this problem without error.

Step 4: Compute X^T

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

Step 5: Combine

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

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

Since $(\Sigma^T\Sigma)^{-1}\Sigma^T = \Sigma^\dagger$ (the pseudoinverse of Σ):

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

Part (c): What happens when we left-multiply X **by** $A = (X^T X)^{-1}X^T$**?**

$$A \cdot X = (X^T X)^{-1}X^T \cdot X = (X^T X)^{-1}(X^T X) = I_n$$

This gives the $n \times n$ identity matrix.

$$\boxed{AX = I_n, \text{ which is why } A = (X^T X)^{-1}X^T \text{ is called the left-inverse}}$$

The left-inverse satisfies $AX = I$ (left multiplication gives identity).

Comparing 3.c to the staff solutions, Claude was able to solve this problem without error.

Part (d): Underdetermined case ($m < n$) - Finding min-norm solution

When $m < n$, we have more unknowns than equations. We want:

$$\min \|\mathbf{w}\|^2 \quad \text{s.t.} \quad X\mathbf{w} = \mathbf{y}$$

Step 1: Set up Lagrangian

$$\mathcal{L}(\mathbf{w}, \lambda) = \|\mathbf{w}\|^2 + \lambda^T(X\mathbf{w} - \mathbf{y})$$

Comparing 3.d to the staff solutions, Claude was able to solve this problem without error.

Step 2: Take derivatives and set to zero

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 2\mathbf{w} + X^T\lambda = 0 \implies \mathbf{w} = -\frac{1}{2}X^T\lambda$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = X\mathbf{w} - \mathbf{y} = 0$$

Step 3: Substitute and solve From the first equation: $\mathbf{w} = -\frac{1}{2}X^T\lambda$

Substitute into the constraint:

$$X\left(-\frac{1}{2}X^T\lambda\right) = \mathbf{y}$$

$$-\frac{1}{2}XX^T\lambda = \mathbf{y}$$

$$\lambda = -2(XX^T)^{-1}\mathbf{y}$$

Step 4: Find optimal w

$$\mathbf{w}^* = -\frac{1}{2}X^T\lambda = -\frac{1}{2}X^T \cdot (-2)(XX^T)^{-1}\mathbf{y}$$

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

Part (e): Plug in SVD and simplify

Step 1: Substitute $X = U\Sigma V^T$

$$XX^T = U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$$

Note: $V^TV = I$ since V is orthonormal.

Step 2: Compute $\Sigma\Sigma^T$ - Σ is $m \times n$ (with $m < n$) - Σ^T is $n \times m$ - $\Sigma\Sigma^T$ is $m \times m$ diagonal with σ_i^2 on the diagonal

Step 3: Compute $(XX^T)^{-1}$

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

Step 4: Compute X^T

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

Step 5: Combine

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

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

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

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

Comparing 3.3 to the staff solutions, Claude was able to solve this problem without error.

Part (f): What happens when we right-multiply X by $B = X^T(XX^T)^{-1}$?

$$X \cdot B = X \cdot X^T(XX^T)^{-1} = (XX^T)(XX^T)^{-1} = I_m$$

This gives the $m \times m$ identity matrix.

$\boxed{XB = I_m}$, which is why $B = X^T(XX^T)^{-1}$ is called the right-inverse

Comparing 3.f to the staff solutions, Claude was able to solve this problem without error.

The right-inverse satisfies $XB = I$ (right multiplication gives identity).

Question 4: The 5 Interpretations of Ridge Regression

Part (a): Perspective 1 - Optimization Problem

Problem: $\arg\min_{\mathbf{w}} \|\mathbf{y} - X\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$

where $X \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^n$.

Step 1: Expand the objective

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

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

Comparing 4.a to the staff solutions, Claude was able to solve this problem without error.

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

Step 2: Take the derivative

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

Using our results from Q2: $\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T X^T X\mathbf{w}) = 2\mathbf{w}^T X^T X$ (since $X^T X$ is symmetric).

Step 3: Set to zero and solve

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

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

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

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

Part (b): Perspective 2 - Shifting Singular Values

Goal: Plug SVD $X = U\Sigma V^T$ into the Ridge solution and analyze.

Step 1: Compute $X^T X$

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

Let $D = \Sigma^T\Sigma$, which is a $d \times d$ diagonal matrix with σ_i^2 on the diagonal.

Step 2: Compute $X^T X + \lambda I$

$$X^T X + \lambda I = VDV^T + \lambda I = VDV^T + \lambda VV^T = V(D + \lambda I)V^T$$

$(D + \lambda I)$ is diagonal with entries $(\sigma_i^2 + \lambda)$.

Step 3: Compute $(X^T X + \lambda I)^{-1}$

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

Comparing 4.b to the staff solutions, Claude was able to solve this problem without error.

where $(D + \lambda I)^{-1}$ is diagonal with entries $\frac{1}{\sigma_i^2 + \lambda}$.

Step 4: Compute the full expression

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

The effective "pseudo-inverse" has entries $\frac{\sigma_i}{\sigma_i^2+\lambda}$ along the diagonal (in the appropriate positions).

Analysis:

When $\sigma_i \ll \lambda$:

$$\frac{\sigma_i}{\sigma_i^2+\lambda} \approx \frac{\sigma_i}{\lambda} \approx 0$$

Small singular values are suppressed to nearly zero.

When $\sigma_i \gg \lambda$:

$$\frac{\sigma_i}{\sigma_i^2+\lambda} \approx \frac{\sigma_i}{\sigma_i^2} = \frac{1}{\sigma_i}$$

Large singular values behave like the ordinary pseudo-inverse.

Ridge regression "shrinks" small singular values toward zero while leaving large ones mostly unchanged.

Part (c): Perspective 3 - MAP Estimation

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

Step 1: Write the posterior By Bayes' rule:

$$P(W|Y = \mathbf{y}) \propto P(Y = \mathbf{y}|W) \cdot P(W)$$

Step 2: Compute the likelihood Given W, we have $Y|W \sim \mathcal{N}(XW, \lambda I)$

$$P(\mathbf{y}|W) \propto \exp\left(-\frac{1}{2\lambda}\|\mathbf{y} - XW\|^2\right)$$

Step 3: Compute the prior

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

Comparing 4.c to the staff solutions, Claude was able to solve this problem without error.

Step 4: Compute the posterior

$$P(W|\mathbf{y}) \propto \exp\left(-\frac{1}{2\lambda}\|\mathbf{y} - XW\|^2 - \frac{1}{2}\|W\|^2\right)$$

Step 5: Find MAP estimate The MAP estimate maximizes the posterior, which is equivalent to minimizing the negative log-posterior:

$$\hat{W}_{MAP} = \arg\min_W \left[\frac{1}{2\lambda}\|\mathbf{y} - XW\|^2 + \frac{1}{2}\|W\|^2\right]$$

Multiplying by λ:

$$\hat{W}_{MAP} = \arg\min_W \left[\frac{1}{2}\|\mathbf{y} - XW\|^2 + \frac{\lambda}{2}\|W\|^2\right]$$

This is equivalent to:

$$\hat{W}_{MAP} = \arg\min_W \left[\|\mathbf{y} - XW\|^2 + \lambda\|W\|^2\right]$$

This is exactly the Ridge Regression objective (1).

Part (d): Perspective 4 - Fake Data

Setup:

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

where $\hat{\mathbf{y}} \in \mathbb{R}^{n+d}$ and $\hat{X} \in \mathbb{R}^{(n+d)\times d}$.

Comparing 4.d to the staff solutions, Claude was able to solve this problem without error.

Step 1: Expand the OLS objective

$$\|\hat{\mathbf{y}} - \hat{X}\mathbf{w}\|_2^2 = \left\|\begin{bmatrix}\mathbf{y} \\ \mathbf{0}_d\end{bmatrix} - \begin{bmatrix}X \\ \sqrt{\lambda} I_d\end{bmatrix}\mathbf{w}\right\|_2^2$$

$$= \left\|\begin{bmatrix}\mathbf{y} - X\mathbf{w} \\ -\sqrt{\lambda}\mathbf{w}\end{bmatrix}\right\|_2^2$$

Step 2: Compute the norm

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

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

This is exactly the Ridge Regression objective (1).

Part (e): Perspective 5 - Fake Features

Setup:

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

We solve:

$$\arg\min_{\eta} \|\eta\|_2^2 \quad \text{s.t.} \quad \check{X}\eta = \mathbf{y}$$

Comparing 4.e to the staff solutions, Claude was able to solve this problem without error.

Step 1: Partition η Let $\eta = \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix}$ where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{z} \in \mathbb{R}^n$.

Step 2: Write the constraint

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

So $\mathbf{z} = \frac{1}{\sqrt{\lambda}}(\mathbf{y} - X\mathbf{w})$.

Step 3: Substitute into the objective

$$\|\eta\|_2^2 = \|\mathbf{w}\|_2^2 + \|\mathbf{z}\|_2^2 = \|\mathbf{w}\|_2^2 + \frac{1}{\lambda}\|\mathbf{y} - X\mathbf{w}\|_2^2$$

Step 4: Minimize over w

$$\min_{\mathbf{w}} \left[\|\mathbf{w}\|_2^2 + \frac{1}{\lambda}\|\mathbf{y} - X\mathbf{w}\|_2^2 \right]$$

Multiplying by λ:

$$\min_{\mathbf{w}} \left[\lambda\|\mathbf{w}\|_2^2 + \|\mathbf{y} - X\mathbf{w}\|_2^2 \right]$$

The first d coordinates of η^* minimize the Ridge objective (1).

Part (f): Showing equivalence of two Ridge formulas

Step 1: Use pseudo-inverse for the min-norm problem (2)

For $\check{X}\eta = \mathbf{y}$, the min-norm solution is:

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

Step 2: Compute $\check{X}\check{X}^T$

Comparing 4.f to the staff solutions, Claude was able to solve this problem without error.

$$\check{X}\check{X}^T = [X \quad \sqrt{\lambda} I_n] \begin{bmatrix} X^T \\ \sqrt{\lambda} I_n \end{bmatrix} = XX^T + \lambda I_n$$

Step 3: Compute $\check{X}^T$

$$\check{X}^T = \begin{bmatrix} X^T \\ \sqrt{\lambda} I_n \end{bmatrix}$$

Step 4: Extract the first d coordinates of η^*

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

The first d coordinates are:

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

Step 5: Show equivalence to standard formula

We need to show:

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

Proof: We verify that both sides give the same result when multiplied on the right by $(XX^T + \lambda I)$ and on the left by $(X^TX + \lambda I)$:

Claim: $(X^TX + \lambda I)^{-1}X^T = X^T(XX^T + \lambda I)^{-1}$

Multiply both sides on the left by $(X^TX + \lambda I)$ and on the right by $(XX^T + \lambda I)$:

LHS: $(X^TX + \lambda I) \cdot (X^TX + \lambda I)^{-1}X^T \cdot (XX^T + \lambda I) = X^T(XX^T + \lambda I)$

$$= X^TXX^T + \lambda X^T$$

RHS: $(X^TX + \lambda I) \cdot X^T(XX^T + \lambda I)^{-1} \cdot (XX^T + \lambda I) = (X^TX + \lambda I)X^T$

$$= X^TXX^T + \lambda X^T$$

Both sides are equal, so:

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

Part (g): What happens when $\lambda \to \infty$?

From the solution $\hat{\mathbf{w}}_r = (X^TX + \lambda I)^{-1}X^T\mathbf{y}$:

As $\lambda \to \infty$:

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

Comparing 4.g to the staff solutions, Claude was able to solve this problem without error.

Therefore:

$$\hat{\mathbf{w}}_r \approx \frac{1}{\lambda} X^T \mathbf{y} \to \mathbf{0}$$

As $\lambda \to \infty, \hat{\mathbf{w}}_r \to \mathbf{0}$. This is why ridge is called "shrinkage" - large λ shrinks weights to zero.

Part (h): What happens when $\lambda \to 0$?

Case 1: Overdetermined system (X is tall, $n > d$)

As $\lambda \to 0$:

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

Comparing 4.h to the staff solutions, Claude was able to solve this problem without error.

This is the ordinary least squares (OLS) solution.

Case 2: Underdetermined system (X is wide, $n < d$)

Using the alternate form: $\hat{\mathbf{w}}_r = X^T (X X^T + \lambda I)^{-1} \mathbf{y}$

As $\lambda \to 0$:

$$\hat{\mathbf{w}}_r \to X^T (X X^T)^{-1} \mathbf{y}$$

This is the minimum-norm solution from Question 3.

In both cases, as $\lambda \to 0$, ridge regression converges to the Moore-Penrose pseudoinverse solution.

Question 5: ReLU Elbow Update under SGD

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

Loss: $\ell = \frac{1}{2} \|\hat{f}(x) - y\|_2^2$

Part (a): Single ReLU Analysis

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

Loss: $\ell(x, y, \phi) = \frac{1}{2} \|\phi(x) - y\|_2^2 = \frac{1}{2} (\phi(x) - y)^2$

Comparing 5.a to the staff solutions, Claude was able to solve this problem without error.

(i) Location of the elbow e:

The elbow is where ϕ transitions from 0 to non-zero, i.e., where $wx + b = 0$:

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

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

(ii) Derivative of loss w.r.t. $\phi(x)$:

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

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

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

Using chain rule: $\frac{\partial \ell}{\partial w} = \frac{d\ell}{d\phi} \cdot \frac{\partial \phi}{\partial w}$

When $\phi(x) > 0$: $\phi(x) = wx + b$, so $\frac{\partial \phi}{\partial w} = x$

When $\phi(x) = 0$: $\frac{\partial \phi}{\partial w} = 0$ (ReLU is in the "off" region)

$$\boxed{\frac{\partial \ell}{\partial w} = \begin{cases} (\phi(x) - y) \cdot x & \text{if } \phi(x) > 0 \\ 0 & \text{if } \phi(x) = 0 \end{cases}}$$

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

Using chain rule: $\frac{\partial \ell}{\partial b} = \frac{d\ell}{d\phi} \cdot \frac{\partial \phi}{\partial b}$

When $\phi(x) > 0$: $\frac{\partial \phi}{\partial b} = 1$

When $\phi(x) = 0$: $\frac{\partial \phi}{\partial b} = 0$

$$\boxed{\frac{\partial \ell}{\partial b} = \begin{cases} \phi(x) - y & \text{if } \phi(x) > 0 \\ 0 & \text{if } \phi(x) = 0 \end{cases}}$$

Comparing 5.b to the staff solutions, Claude was able to solve this problem without error.

Part (b): Gradient descent when $\phi(x) - y = 1$

The prediction is 1 unit above target. Update rule: $p' = p - \lambda \nabla_p \ell$

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

When $\phi(x) = 0$, we're in the "off" region of the ReLU. - $\frac{\partial \ell}{\partial w} = 0$ - $\frac{\partial \ell}{\partial b} = 0$

No change to slope or elbow. The gradients are zero when the ReLU is inactive.

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

Given $\phi(x) - y = 1$: - $\frac{\partial \ell}{\partial w} = 1 \cdot x = x > 0$ - $\frac{\partial \ell}{\partial b} = 1$

Updates: - $w' = w - \lambda x$ (slope decreases since $x > 0$) - $b' = b - \lambda$ (bias decreases)

New elbow location:

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

Let's analyze: with $w > 0$, $x > 0$, and $\phi(x) = wx + b > 0$, the elbow was at $e = -b/w$.

Since both w and b decrease, we need to compare the ratios. The elbow moves based on the relative changes.

Slope decreases. The elbow location changes depending on $\frac{b}{w}$ vs $\frac{1}{x}$.

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

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

Updates: - $w' = w - \lambda x = w + \lambda |x|$ (slope increases since $x < 0$) - $b' = b - \lambda$ (bias decreases)

New elbow:

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

Since we're at $x < 0$ but $\phi(x) > 0$, the elbow must be to the left of x.

Slope increases, bias decreases. Elbow moves right (toward more negative x values becoming inactive).

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

For $\phi(x) = wx + b > 0$ with $w < 0$ and $x > 0$: we need $b > |w|x$.

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

Updates: - $w' = w - \lambda x$ (becomes more negative, slope magnitude increases) - $b' = b - \lambda$ (bias decreases)

Slope becomes more negative (steeper downward). Elbow moves based on relative changes in b and w.

Diagrams Description: For each case where $\phi(x) > 0$: - Draw the original ReLU with elbow marked - Show the training point (x, y) with $\phi(x) = y + 1$ (prediction above target) - Draw arrows showing the function moving downward toward the target - Indicate elbow movement direction

Note that Claude did not or was unable to draw the necesary diagrams but instead described them.

Part (c): Elbow location for the i-th ReLU in full network

The i-th ReLU receives input: $z_i = W_i^{(1)} x + b_i$

where $W_i^{(1)}$ is the i-th row of $W^{(1)}$ (a scalar for 1D input).

The ReLU activates when $z_i > 0$, i.e., when $W_i^{(1)} x + b_i > 0$.

The elbow (transition point) is where $W_i^{(1)} x + b_i = 0$:

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

Part (d): New elbow location after SGD update

Step 1: Compute gradients for the full network

Let $z_i = W_i^{(1)} x + b_i$ and $a_i = \text{ReLU}(z_i)$.

Output: $\hat{f}(x) = \sum_j W_j^{(2)} a_j$

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

Step 2: Backpropagation

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

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

$$\frac{\partial \ell}{\partial z_i} = (\hat{f}(x) - y) W_i^{(2)} \cdot \mathbf{1}_{z_i > 0}$$

$$\frac{\partial \ell}{\partial W_i^{(1)}} = \frac{\partial \ell}{\partial z_i} \cdot x = (\hat{f}(x) - y) W_i^{(2)} \cdot \mathbf{1}_{z_i > 0} \cdot x$$

$$\frac{\partial \ell}{\partial b_i} = \frac{\partial \ell}{\partial z_i} = (\hat{f}(x) - y) W_i^{(2)} \cdot \mathbf{1}_{z_i > 0}$$

Step 3: SGD updates

Let $\delta = (\hat{f}(x) - y) W_i^{(2)} \cdot \mathbf{1}_{z_i > 0}$

$$(W_i^{(1)})' = W_i^{(1)} - \lambda \delta x$$
$$b_i' = b_i - \lambda \delta$$

Step 4: New elbow location

$$e_i' = -\frac{b_i'}{(W_i^{(1)})'} = -\frac{b_i - \lambda \delta}{W_i^{(1)} - \lambda \delta x}$$

If $z_i \leq 0$ (ReLU inactive), then $\delta = 0$ and $e_i' = e_i$ (no change).

If $z_i > 0$ (ReLU active):

$$\boxed{e_i' = -\frac{b_i - \lambda(\hat{f}(x) - y) W_i^{(2)}}{W_i^{(1)} - \lambda(\hat{f}(x) - y) W_i^{(2)} x}}$$

Or equivalently, with $\delta = (\hat{f}(x) - y) W_i^{(2)}$:

$$\boxed{e_i' = \frac{-b_i + \lambda \delta}{W_i^{(1)} - \lambda \delta x} = \frac{e_i W_i^{(1)} + \lambda \delta}{W_i^{(1)} - \lambda \delta x}}$$

Summary

This homework covers fundamental concepts essential for deep learning:

1. **Vector Calculus (Q2):** Forms the basis for understanding gradient computation in neural networks.
2. **SVD and Linear Algebra (Q3):** Provides insight into how linear systems are solved, which underlies many optimization algorithms.

3. **Ridge Regression (Q4):** Shows multiple perspectives on regularization, a critical concept for preventing overfitting.
4. **ReLU and SGD (Q5):** Demonstrates how gradient descent updates affect the geometry of neural network activation functions.

Overall I was very impressed with Claude's work. It solved each question correctly in one-shot and in many cases had more detailed explanations of it's work and steps than the staff solutions.