

Me: [Image of problem 1] Solve problem 1 a

Gemini 3 Pro: Show Thinking

### Initiating Problem Breakdown

I'm currently focused on the initial request, which involves understanding the user's need to solve part (a) of the optimization problem depicted in the image. I've broken down the inputs – the image and its embedded problem – and I'm pinpointing the core task: minimizing the given objective function using an appropriate optimization method. My approach will be to extract relevant information.

### Calculating Analytical Solution

I've simplified the problem to a quadratic optimization with a specific Euclidean norm. Now I'm diving into the math. My objective is to derive an analytical solution for the optimal update,  $\Delta\theta$  (which the problem calls  $u$ ), and then see what known optimizer this corresponds to. The objective function is  $J(\Delta\theta) = g^T \Delta\theta + \frac{1}{\alpha} \|\Delta\theta\|_2^2$ , where  $g$  is the gradient and  $\alpha$  is a hyperparameter. I'm taking the gradient.

### Defining the Update Rule

I've just finished the derivation, and it's quite neat. We can now explicitly define the optimal update rule. After taking the gradient and solving, the expression emerges as  $\Delta\theta = -\frac{\alpha}{2} g$ . I've interpreted this as a gradient descent step. If we define the learning rate  $\eta = \frac{\alpha}{2}$ , the update rule elegantly becomes  $\Delta\theta = -\eta \nabla f(\theta)$ . This is a well-known update rule from optimization.

### Analyzing Problem Components

I'm now structuring the explanation for the user. I'll begin with the objective function and proceed step-by-step through the derivation of the optimal update rule. I intend to explicitly show the process, including vector calculus rules, as I move toward the identification of the optimizer. Finally, I'll ensure my notation aligns with the problem statement.

### Confirming the Optimizer

I'm solidifying the derivation by carefully extracting each mathematical step. I now need to write out the formal response. The goal is to provide a clear and organized breakdown for part (a). I'm focusing on crafting a structure that starts with the objective function and then logically walks through the key calculations to reach the conclusion. I'm paying close attention to notation to ensure it aligns precisely with the given problem. After deriving the optimal

update ( $\$u\$$ ), I will link it to the definition of a known optimizer.

## Verifying Solution Structure

I'm now fully immersed in organizing the response. I've re-examined the problem to make sure the assumptions are consistent, especially regarding the squared norm and the positive nature of alpha. I have confirmed that the squared norm is the correct distance function to use and the optimization is unconstrained. I'm structuring the explanation starting from the objective function. Then, after calculating the gradient, I'll arrive at the update rule for theta.

Based on the image provided, here is the step-by-step derivation for the analytical solution to part (a).

## Objective

We want to find the vector  $\$u$$  that minimizes the following objective function with respect to  $\Delta\theta$ :

$$J(\Delta\theta) = g^T \Delta\theta + \frac{1}{\alpha} \|\Delta\theta\|_2^2$$

Here:

- $\$g = \nabla f(\theta)$  is the gradient vector.
- $\|\Delta\theta\|_2^2 = \Delta\theta^T \Delta\theta$  is the squared Euclidean norm (L2 norm).
- $\alpha$  is a scalar weighting parameter.

## Derivation

1. Expand the objective function:

Using matrix notation, we can rewrite the squared L2 norm:

$$J(\Delta\theta) = g^T \Delta\theta + \frac{1}{\alpha} (\Delta\theta^T \Delta\theta)$$

2. Take the derivative with respect to  $\Delta\theta$ :

We differentiate the function to find the critical points.

- The derivative of  $g^T \Delta\theta$  with respect to  $\Delta\theta$  is simply  $g$ .
- The derivative of  $\Delta\theta^T \Delta\theta$  with respect to  $\Delta\theta$  is  $2\Delta\theta$ .

So, the gradient of the objective function is:

$$\nabla_{\Delta\theta} J = g + \frac{1}{\alpha}(2\Delta\theta)$$

3. Set the derivative to zero to find the minimum:

To minimize the objective, we set the gradient equal to the zero vector:

$$g + \frac{2}{\alpha}\Delta\theta = 0$$

4. Solve for  $\Delta\theta$  (which is our optimal update  $u$ ):

$$\frac{2}{\alpha}\Delta\theta = -g$$

Multiply both sides by  $\frac{\alpha}{2}$ :

$$\Delta\theta = -\frac{\alpha}{2}g$$

Since  $u$  is defined as the  $\text{argmin}$  (the value of  $\Delta\theta$  that minimizes the function), we have:

$$u = -\frac{\alpha}{2}g$$

## Final Answer

Analytical Solution:

$$u = -\frac{\alpha}{2}\nabla f(\theta)$$

(Note: I substituted  $g$  back with  $\nabla f(\theta)$  for clarity).

Optimizer Recovered:

This recovers Gradient Descent (or Standard Gradient Descent).

**Reasoning:** The solution shows that the update step  $u$  is directly proportional to the negative of the gradient. In standard Gradient Descent, the update rule is  $\Delta\theta = -\eta \nabla f(\theta)$ , where  $\eta$  is the learning rate. In this derivation, the term

$\frac{\alpha}{2}$  acts effectively as the learning rate  $\eta$ .

Me: [Image of problem 1 part b] Solve

Gemini 3 pro: Show Thinking

### Tackling the Problem

I'm now zeroing in on part (b) of the image's optimization challenge. The analytical solution for 'u' within the specified objective function is the main quest. My plan: isolate 'u' using mathematical manipulation, then check my work.

### Decomposing the Equation

I've been breaking down the objective function, identifying the linear and regularization terms. Currently, I'm setting M equal to the infinite norm and reformulating the minimization problem, imposing constraints on each theta. My current tactic is to determine the optimal  $\Delta\theta_i$  for a fixed M to further clarify.

### Finding the Minimum

My focus has shifted to minimizing the objective function with respect to  $M$ . I've taken the derivative and found an expression for  $M$  that makes the derivative zero. Solving for  $M$ , it hinges on the 1-norm of  $g$  and the regularization parameter,  $\alpha$ . Specifically,  $M$  equals  $\alpha$  over 2 times the 1-norm of  $g$ . I believe I've pinpointed the key scalar parameter now and how it affects the overall outcome.

### Confirming the Approach

I've plugged the value of  $M$  back into the  $\Delta\theta_i$  equation, simplifying it into a vector form:  $u = -\frac{\alpha}{2} \|g\|_1 \text{sign}(g)$ . My next step was to relate this to common optimizer formulations. It does resemble a steepest descent using the  $L_\infty$  norm, where the direction hinges entirely on the gradient signs. The step size scales with the gradient's 1-norm.

### Relating to Known Optimizers

I'm now linking my solution to common optimization techniques. It's similar to Signed Gradient Descent, where the update's direction is the gradient's sign. My current work is focused on connecting this result to standard optimizer classes. It's not exactly Adam, nor RMSprop. My direction now is to compare the approach with established optimizer families, specifically those related to gradient signs.

## Pinpointing Optimizer Equivalence

I'm now zeroing in on the equivalent optimizer class. It's clear my solution resembles SignSGD, where updates are proportional to the gradient's sign. The adaptive learning rate,  $\alpha/2 * \|g\|_1$ , distinguishes it. It's not standard Coordinate Descent since all parameters update simultaneously. My analysis suggests this is steepest descent in the L-infinity norm, a special case of SignSGD, or simply "Signed Gradient Descent". This seems to align perfectly with the derived update rule.

## Interpreting the Results

I'm now exploring alternative interpretations of the solution, specifically considering L1 regularization on the update step, which doesn't seem to fit the derived update rule. The L-infinity penalty, however, leads directly to the observation that the update is dense, aligning with SignSGD. I've confirmed the maths again and am satisfied.

## Finalizing Optimizer Choice

I'm now revisiting the nuances of the identified optimizer. While the derived solution is precisely  $u = -\frac{\alpha}{2} \text{sign}(g)$ , the question specifically asks for an optimizer. The key here is the *direction* of the update, which is defined by the sign. Hence, the most fitting name is **SignSGD**, as the scaling factor doesn't alter this core aspect. The  $L_\infty$  penalty confirms this interpretation.

## Choosing the Right Name

My thinking has progressed to the exact name of the optimizer. Initially, I thought about Rprop and Adam, but the core update rule is the gradient sign, aligning precisely with SignSGD. Also, the geometric interpretation is *Steepest Descent in the  $L_\infty$  norm*. However, the common name I see is SignSGD. I'll need to write the final answer.

## Finalizing Optimizer Choice

I'm now revisiting the potential for Block Coordinate Descent. My initial focus was on SignSGD, but the L1 versus Linf penalty perspective changes things. L1 regularization on the update leads to sparse updates, which corresponds to Coordinate Descent (Steepest). Conversely, Linf regularization results in dense updates, aligning with SignSGD. The math checks out. I've reconfirmed the approach in the context of the L-p norm and the dual norm.

## Analyzing the Steps

I've carefully revisited each step, expanding terms, optimizing the direction, and substituting back. The objective function became a scalar function  $J(m)$ , which was then differentiated and set to zero. I verified the final solution,  $u = -\frac{\alpha}{2} \text{sign}(g)$

again, confirming that each step is mathematically sound and consistent.

### **Formalizing the Update**

I'm now writing up the formal proof. I'm focusing on the mathematical justification behind identifying the optimizer as Signed Gradient Descent, or SignSGD. This involves re-deriving the update rule within a standard gradient descent framework and highlighting the key role of the gradient's sign. I will explicitly contrast this approach with adaptive optimizers to reinforce the choice. I need to make sure the presentation is clear and concise, emphasizing the mathematical equivalency.

### **Formalizing the Solution**

I'm solidifying my approach and focusing on conveying the result succinctly. I'll emphasize that the update rule translates directly to Signed Gradient Descent (SignSGD). My presentation will highlight how this update derives from an L-infinity penalty, and the significance of the gradient sign.

### **Verifying the Connection**

My current thinking revolves around refining the exact optimizer class. I'm solidifying the link between the L-infinity norm constraint and the resulting optimizer. Steepest descent in the L1 norm corresponds to Coordinate Descent, while steepest descent in the L-infinity norm translates to Sign Descent. The squared infinity norm in (b) points directly to Signed Gradient Descent, confirming the direction.

### **Connecting the Concepts**

I'm now writing up the formal proof. My focus is the clear relationship between the gradient and the chosen optimizer. The  $L_{\infty}$  penalty results in the update direction being based on the gradient's sign. The analytical solution is confirmed. The most appropriate name is SignSGD.

Here is the step-by-step analytical derivation for part (b).

### **Objective**

We want to find the update vector  $\Delta\theta$  (denoted as  $\Delta\theta$ ) that minimizes:

$$J(\Delta\theta) = g^T \Delta\theta + \frac{1}{\alpha} \|\Delta\theta\|_\infty^2$$

Where:

- $\mathbf{g} = \nabla f(\theta)$
- $\|\Delta\theta\|_\infty = \max_i |\Delta\theta_i|$  (the maximum absolute value among all components).

## Derivation

1. Define a scalar for the norm:

Let  $m = \|\Delta\theta\|_\infty$ . By definition, this means for every component  $i$ :

$$|\Delta\theta_i| \leq m \implies -m \leq \Delta\theta_i \leq m$$

We can rewrite the objective function in terms of  $m$ :

$$J = \sum_i (g_i \Delta\theta_i) + \frac{1}{\alpha} m^2$$

2. Minimize the linear term for a fixed  $m$ :

To minimize the sum  $\sum g_i \Delta\theta_i$ , we should choose each  $\Delta\theta_i$  such that it has the opposite sign of  $g_i$  and the maximum possible magnitude allowed by the constraint  $m$ .

- If  $g_i$  is positive, we want  $\Delta\theta_i = -m$ .
- If  $g_i$  is negative, we want  $\Delta\theta_i = +m$ .

Mathematically, this can be written using the sign function:

$$\Delta\theta_i = -m \cdot \text{sign}(g_i)$$

Now, substitute this back into the linear term:

$$\sum_i g_i \Delta\theta_i = \sum_i g_i (-m \cdot \text{sign}(g_i)) = -m \sum_i g_i \text{sign}(g_i)$$

Since  $x \cdot \text{sign}(x) = |x|$ , this simplifies to:

$$-m \sum_i |g_i| = -m \|g\|_1$$

(Note:  $\|g\|_1$  is the L1 norm of the gradient).

3. Solve for  $m$ :

Now our objective function depends only on the scalar  $m$ :

$$J(m) = -m\|g\|_1 + \frac{1}{\alpha}m^2$$

To find the optimal  $m$ , take the derivative with respect to  $m$  and set it to 0:

$$\frac{dJ}{dm} = -\|g\|_1 + \frac{2}{\alpha}m = 0$$

$$\frac{2}{\alpha}m = \|g\|_1$$

$$m = \frac{\alpha}{2}\|g\|_1$$

4. Construct the final solution  $u$ :

Substitute the optimal  $m$  back into our expression for  $\Delta\theta_i$ :

$$u_i = -\left(\frac{\alpha}{2}\|g\|_1\right)\text{sign}(g_i)$$

In vector notation:

$$u = -\frac{\alpha}{2}\|\nabla f(\theta)\|_1 \cdot \text{sign}(\nabla f(\theta))$$


---

## Final Answer

Analytical Solution:

$$u = -\frac{\alpha}{2}\|\nabla f(\theta)\|_1 \cdot \text{sign}(\nabla f(\theta))$$

Optimizer Recovered:

This corresponds to Signed Gradient Descent (often referred to as SignSGD or Steepest Descent in the  $L_\infty$  Norm).

Reasoning:

Unlike standard Gradient Descent which scales updates based on the magnitude of the gradient, this optimizer updates every parameter by the same step size (proportional to the global L1 norm of the gradient), with the direction determined solely by the sign of the gradient element-wise.

