Layer norm backward

Ten problems on the backward pass of layer normalization: the gradients of the mean and variance, the centering projection, the Jacobian of x̂, the input gradient and why it is orthogonal to 1 and, up to ε, to x̂, the batched version and RMSNorm, with worked solutions and the mistakes that treat μ and σ as constants.

BEFORE YOU START

Layer normalization sits in every transformer block: it subtracts each token's mean, divides by its standard deviation, then rescales and shifts with learned γ and β. The backward pass through γ and β is one line each; the backward pass to the input is where the work is, because the mean and the standard deviation are themselves functions of every input feature. These ten problems build that gradient from its pieces, show why it can only push the input in certain directions, and then batch it and strip it down to RMSNorm. The four mistakes are the ones that give a plausible-looking gradient: μ and σ held constant, γ's gradient read off the output, a $1/\sigma$ applied to one term, and a shared parameter's gradient left unsummed.

- The conventions are those of the previous pages: vectors are columns, Jacobians are in numerator layout, a gradient has the shape of the variable it is taken with respect to, and for a scalar L of u with u a function of x, $\nabla_x L = (\partial u/\partial x)^\top \nabla_u L$.
- One token's features are $x \in \mathbb{R}^n$. Its mean is $\mu = \frac{1}{n}\sum_i x_i$, its variance $v = \frac{1}{n}\sum_i (x_i - \mu)^2$, and $\sigma = \sqrt{v + \epsilon}$; on this page σ is this standard deviation, not the sigmoid.
- ϵ is a small constant inside the square root; keep it, it is what stops the Jacobian blowing up on a constant row.
- The normalized input is $\hat{x} = (x - \mu\mathbf{1})/\sigma$ and the output is $y = \gamma \odot \hat{x} + \beta$, with learned $\gamma, \beta \in \mathbb{R}^n$. $\mathbf{1}$ is the all-ones vector and $\odot$ the elementwise product.
- L is a scalar loss that depends on x, γ and β only through y. The upstream gradient is $g = \nabla_y L$, and $g' = \gamma \odot g$; Problem 5 shows $g' = \nabla_{\hat{x}} L$.
- $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, and $\text{mean}(u) = \frac{1}{n}\mathbf{1}^\top u$ for any $u \in \mathbb{R}^n$, so $\text{mean}(g' \odot \hat{x}) = \frac{1}{n}\hat{x}^\top g'$.
- Batched: rows are tokens. $X \in \mathbb{R}^{N \times n}$ and the normalization is per row: each row gets its own μ, v and σ, while γ and β are shared by all rows.

Builds on: Matrix calculus conventions, Jacobians and the chain rule

Problems

1. Compute $\nabla_x \mu$. difficulty ·

2. Compute $\nabla_x v$. Why does the dependence of μ on x not add a term? difficulty ··

3. Let $c = x - \mu \mathbf{1}$. Compute $\partial c / \partial x$, and show it is symmetric, idempotent and sends $\mathbf{1}$ to 0. difficulty ··

4. Compute $\nabla_\gamma L$ and $\nabla_\beta L$ in terms of g and $\hat{x}$. difficulty ··

5. Compute $\nabla_{\hat{x}} L$. difficulty ··

6. Compute $\partial \hat{x} / \partial x$ as a matrix. difficulty ···

7. Compute $\nabla_x L$ in terms of $g' = \gamma \odot g$, $\hat{x}$ and σ, using only vector operations (no $n \times n$ matrix). difficulty ···

8. Show that $\mathbf{1}^\top \nabla_x L = 0$, and compute $\hat{x}^\top \nabla_x L$; when is it 0? What does that say about which directions of x the loss can push on? difficulty ··

9. Batched: $X \in \mathbb{R}^{N \times n}$ with rows normalized independently, $G = \nabla_Y L$. Write $\nabla_X L$, $\nabla_\gamma L$ and $\nabla_\beta L$. difficulty ···

10. RMSNorm drops the mean: $y = \gamma \odot x / r$ with $r = \sqrt{\frac{1}{n}\sum_i x_i^2 + \epsilon}$. Compute $\nabla_x L$. difficulty ···

Answers

1. $\nabla_x \mu = \frac{1}{n}\mathbf{1}$
2. $\nabla_x v = \frac{2}{n}(x - \mu\mathbf{1})$
3. $\partial c/\partial x = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top = P$; $P^\top = P$, $P^2 = P$, $P\mathbf{1} = 0$
4. $\nabla_\gamma L = g \odot \hat{x}$; $\nabla_\beta L = g$
5. $\nabla_{\hat{x}} L = \gamma \odot g = g'$
6. $\partial \hat{x}/\partial x = \frac{1}{\sigma}\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\hat{x}\hat{x}^\top\right)$
7. $\nabla_x L = \frac{1}{\sigma}\left(g' - \text{mean}(g')\mathbf{1} - \hat{x}\,\text{mean}(g' \odot \hat{x})\right)$
8. $\mathbf{1}^\top \nabla_x L = 0$ exactly, and $\hat{x}^\top \nabla_x L = \dfrac{1}{\sigma}\,\dfrac{\epsilon}{v+\epsilon}\,\hat{x}^\top g'$, which is 0 when $\epsilon = 0$ or when $\hat{x}^\top g' = 0$, and of order ϵ otherwise: the loss cannot move x along $\mathbf{1}$ (a shift), and can move it along $\hat{x}$ (a rescale) only through the ϵ term
9. $\nabla_X L$ is Problem 7 applied to each row; $\nabla_\gamma L = (G \odot \hat{X})^\top \mathbf{1}$; $\nabla_\beta L = G^\top \mathbf{1}$
10. $\nabla_x L = \frac{1}{r}\left(g' - \hat{x}\,\text{mean}(g' \odot \hat{x})\right)$ with $\hat{x} = x/r$

Worked solutions

Problem 1

> Compute $\nabla_x \mu$.

$\partial \mu / \partial x_j = \frac{1}{n}$ for every j.	$\mu = \frac{1}{n}\sum_i x_i$ contains x_j once, with coefficient $\frac{1}{n}$.
$\nabla_x \mu = \frac{1}{n}\mathbf{1}$	$n \times 1$, the shape of x. Every feature moves the mean equally, so in Problem 3 every entry of $\partial(\mu\mathbf{1})/\partial x$ is $\frac{1}{n}$.

Problem 2

> Compute $\nabla_x v$. Why does the dependence of μ on x not add a term?

$\dfrac{\partial v}{\partial x_j} = \dfrac{1}{n}\sum_i 2(x_i - \mu)\Big(\dfrac{\partial x_i}{\partial x_j} - \dfrac{\partial \mu}{\partial x_j}\Big).$	Chain rule on each square, keeping μ as the function of x that it is.
$= \frac{2}{n}(x_j - \mu) - \frac{2}{n}\sum_i (x_i - \mu)\,\dfrac{\partial \mu}{\partial x_j}.$	$\partial x_i / \partial x_j$ is 1 for $i = j$ and 0 otherwise, which picks out the direct term; $\partial \mu / \partial x_j$ does not depend on i, so it can stay inside or come out of the sum.
$\sum_i (x_i - \mu) = \sum_i x_i - n\mu = 0$, so the second term vanishes.	$n\mu = \sum_i x_i$ by the definition of μ: deviations from the mean sum to zero.
$\nabla_x v = \frac{2}{n}(x - \mu\mathbf{1})$	$n \times 1$. The μ term is not missing by accident: μ is the value of m that minimizes $\frac{1}{n}\sum_i (x_i - m)^2$, so the derivative of that sum with respect to m is zero at $m = \mu$, and a small change in μ changes v only to second order.

Problem 3

> Let $c = x - \mu\mathbf{1}$. Compute $\partial c/\partial x$, and show it is symmetric, idempotent and sends $\mathbf{1}$ to 0.

$\partial c/\partial x = I - \mathbf{1}\,(\nabla_x \mu)^\top$.	$\partial x/\partial x = I$, and $\mu\mathbf{1}$ has entry i equal to μ, so row i of its Jacobian is $(\nabla_x \mu)^\top$ for every i.
$\mathbf{1}\,(\nabla_x \mu)^\top = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, an $n \times n$ matrix with every entry $\frac{1}{n}$.	Problem 1.
$P^\top = I^\top - \frac{1}{n}(\mathbf{1}\mathbf{1}^\top)^\top = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top = P$.	$(ab^\top)^\top = ba^\top$, and here $a = b = \mathbf{1}$.
$P^2 = I - \frac{2}{n}\mathbf{1}\mathbf{1}^\top + \frac{1}{n^2}\mathbf{1}(\mathbf{1}^\top\mathbf{1})\mathbf{1}^\top = I - \frac{2}{n}\mathbf{1}\mathbf{1}^\top + \frac{1}{n}\mathbf{1}\mathbf{1}^\top = P$.	Expand the product; $\mathbf{1}^\top\mathbf{1} = n$, so the last term is $\frac{n}{n^2}\mathbf{1}\mathbf{1}^\top$.
$P\mathbf{1} = \mathbf{1} - \frac{1}{n}\mathbf{1}(\mathbf{1}^\top\mathbf{1}) = \mathbf{1} - \mathbf{1} = 0$.	Again $\mathbf{1}^\top\mathbf{1} = n$.
$\partial c/\partial x = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top = P$; $P^\top = P$, $P^2 = P$, $P\mathbf{1} = 0$	$n \times n$. c is linear in x, so $c = Px$: P is the orthogonal projection onto the vectors whose entries sum to zero. Centering twice changes nothing ($P^2 = P$), and a constant vector centers to zero ($P\mathbf{1} = 0$).

Problem 4

> Compute $\nabla_\gamma L$ and $\nabla_\beta L$ in terms of g and $\hat{x}$.

$y_i = \gamma_i \hat{x}_i + \beta_i$.	$\odot$ is elementwise, and $\hat{x}$ does not depend on γ or β.
γ_i and β_i appear only in y_i, with coefficients $\hat{x}_i$ and 1.	Entry i of γ and of β scales and shifts only feature i.
$\partial L/\partial \gamma_i = g_i \hat{x}_i$ and $\partial L/\partial \beta_i = g_i$.	L depends on γ and β only through y; by step 2 only the y_i term of the chain rule is nonzero, and $\partial L/\partial y_i = g_i$.

$\nabla_\gamma L = g \odot \hat{x};\ \nabla_\beta L = g$	Both $n \times 1$, the shapes of γ and β.

Problem 5

> Compute $\nabla_{\hat{x}} L$.

$\partial y / \partial \hat{x} = \operatorname{diag}(\gamma),\ n \times n.$	$y_i = \gamma_i \hat{x}_i + \beta_i$ contains only $\hat{x}_i$, with coefficient γ_i.
$\nabla_{\hat{x}} L = \operatorname{diag}(\gamma)^\top g = \operatorname{diag}(\gamma)\, g.$	L depends on $\hat{x}$ only through y, and a diagonal matrix is its own transpose.
$\nabla_{\hat{x}} L = \gamma \odot g = g'$	A diagonal matrix times a vector scales entry i by γ_i, so the $n \times n$ matrix is never built. From here on the backward pass only needs g'.

Problem 6

> Compute $\partial \hat{x} / \partial x$ as a matrix.

$\hat{x}_i = c_i / \sigma$, so $\dfrac{\partial \hat{x}_i}{\partial x_j} = \dfrac{1}{\sigma}\dfrac{\partial c_i}{\partial x_j} - \dfrac{c_i}{\sigma^2}\dfrac{\partial \sigma}{\partial x_j}.$	Quotient rule, with σ a function of x like c.
$\nabla_x \sigma = \dfrac{1}{2\sigma}\nabla_x v = \dfrac{1}{2\sigma} \cdot \dfrac{2}{n}\, c = \dfrac{1}{\sigma} \cdot \dfrac{1}{n}(x - \mu \mathbf{1}) = \dfrac{1}{n}\hat{x}.$	$\sigma = \sqrt{v + \epsilon}$ with ϵ constant, so $d\sigma/dv = 1/(2\sqrt{v+\epsilon}) = 1/(2\sigma)$; $\nabla_x v$ is Problem 2. This is where ϵ enters: through σ only.
$\partial \hat{x} / \partial x = \dfrac{1}{\sigma} P - \dfrac{1}{\sigma^2}\, c\, (\nabla_x \sigma)^\top = \dfrac{1}{\sigma} P - \dfrac{1}{\sigma} \cdot \dfrac{1}{n}\, \hat{x}\hat{x}^\top.$	Step 1 as a matrix: $\partial c / \partial x = P$ (Problem 3), and entry (i, j) of $c\,(\nabla_x \sigma)^\top$ is $c_i\, \partial \sigma / \partial x_j$. Then $c/\sigma = \hat{x}$ and step 2.
$\partial \hat{x} / \partial x = \frac{1}{\sigma}\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\hat{x}\hat{x}^\top\right)$	$n \times n$ and symmetric. It has the shape of the Jacobians page's Problem 9, the Jacobian of $x/\lVert x \rVert$, with a centering added and σ in place of the norm. On a constant row $v = 0$ and $\hat{x} = 0$, so the Jacobian is $P/\sqrt{\epsilon}$: large but finite, where with $\epsilon = 0$ it would divide by zero.

Problem 7

> Compute $\nabla_x L$ in terms of $g' = \gamma \odot g$, $\hat{x}$ and σ, using only vector operations (no $n \times n$ matrix).

$\nabla_x L = (\partial \hat{x}/\partial x)^\top g' = (\partial \hat{x}/\partial x)\, g'.$	L depends on x only through $\hat{x}$ (through y), $\nabla_{\hat{x}} L = g'$ is Problem 5, and the Jacobian of Problem 6 is symmetric.
$= \frac{1}{\sigma}\big(g' - \frac{1}{n}\mathbf{1}(\mathbf{1}^\top g') - \frac{1}{n}\hat{x}(\hat{x}^\top g')\big).$	Multiply each term of Problem 6 by g' and regroup: $(\mathbf{1}\mathbf{1}^\top)g' = \mathbf{1}(\mathbf{1}^\top g')$ and $(\hat{x}\hat{x}^\top)g' = \hat{x}(\hat{x}^\top g')$, each a vector times a scalar.
$\frac{1}{n}\mathbf{1}^\top g' = \mathrm{mean}(g')$ and $\frac{1}{n}\hat{x}^\top g' = \mathrm{mean}(g' \odot \hat{x}).$	Both are the definition of mean; the second uses $\hat{x}^\top g' = \sum_i \hat{x}_i g'_i = \mathbf{1}^\top(g' \odot \hat{x})$.
$\nabla_x L = \frac{1}{\sigma}\big(g' - \mathrm{mean}(g')\mathbf{1} - \hat{x}\,\mathrm{mean}(g' \odot \hat{x})\big)$	$n \times 1$, the shape of x. It needs two means and a few elementwise operations, $O(n)$ work, against $O(n^2)$ to build and apply the Jacobian. σ and $\hat{x}$ are saved from the forward pass.

Problem 8

> Show that $\mathbf{1}^\top \nabla_x L = 0$, and compute $\hat{x}^\top \nabla_x L$; when is it 0? What does that say about which directions of x the loss can push on?

$\mathbf{1}^\top \hat{x} = \frac{1}{\sigma}\mathbf{1}^\top c = \frac{1}{\sigma}\sum_i (x_i - \mu) = 0$, exactly.	Deviations from the mean sum to zero (Problem 2, step 3), whatever ϵ is.
$\hat{x}^\top \hat{x} = \dfrac{c^\top c}{\sigma^2} = \dfrac{nv}{v + \epsilon}.$	$c^\top c = \sum_i (x_i - \mu)^2 = nv$ and $\sigma^2 = v + \epsilon$. This equals n only when $\epsilon = 0$.
$\mathbf{1}^\top \nabla_x L = \frac{1}{\sigma}\big(\mathbf{1}^\top g' - n\,\mathrm{mean}(g') - (\mathbf{1}^\top \hat{x})\,\mathrm{mean}(g' \odot \hat{x})\big) = \frac{1}{\sigma}(\mathbf{1}^\top g' - \mathbf{1}^\top g' - 0) = 0.$	Problem 7, with $\mathbf{1}^\top \mathbf{1} = n$, $n\,\mathrm{mean}(g') = \mathbf{1}^\top g'$, and step 1.
$\hat{x}^\top \nabla_x L = \frac{1}{\sigma}\big(\hat{x}^\top g' - (\hat{x}^\top \mathbf{1})\,\mathrm{mean}(g') - (\hat{x}^\top \hat{x})\,\frac{1}{n}\hat{x}^\top g'\big).$	Problem 7 again, with $\mathrm{mean}(g' \odot \hat{x}) = \frac{1}{n}\hat{x}^\top g'$.

$= \frac{1}{\sigma}\,\hat{x}^\top g'\Big(1 - \dfrac{v}{v+\epsilon}\Big) = \dfrac{1}{\sigma}\,\dfrac{\epsilon}{v+\epsilon}\,\hat{x}^\top g'.$	The middle term is 0 by step 1; by step 2 the last is $\frac{v}{v+\epsilon}\hat{x}^\top g'$, and $1 - \frac{v}{v+\epsilon} = \frac{\epsilon}{v+\epsilon}$. The ϵ survives because $\|\hat{x}\|^2$ falls just short of n.
$\mathbf{1}^\top \nabla_x L = 0$ exactly, and $\hat{x}^\top \nabla_x L = \dfrac{1}{\sigma}\,\dfrac{\epsilon}{v+\epsilon}\,\hat{x}^\top g'$, which is 0 when $\epsilon = 0$ or when $\hat{x}^\top g' = 0$, and of order ϵ otherwise: the loss cannot move x along 1 (a shift), and can move it along $\hat{x}$ (a rescale) only through the ϵ term	Adding $a\mathbf{1}$ to x adds a to μ and leaves c, v and $\hat{x}$ unchanged, so L cannot change. Scaling c by $1 + a$ scales v by $(1+a)^2$, and $\hat{x} = c/\sqrt{v+\epsilon}$ would be unchanged if ϵ were 0. With $\epsilon = 10^{-5}$ and v near 1 the factor $\epsilon/(v+\epsilon)$ is about 10^{-5}; it matters only on rows whose variance is near ϵ. So a gradient step on x never changes its mean and barely changes its spread: layer norm's output does not see the mean at all, and sees the spread only through ϵ.

Problem 9

> Batched: $X \in \mathbb{R}^{N \times n}$ with rows normalized independently, $G = \nabla_Y L$. Write $\nabla_X L$, $\nabla_\gamma L$ and $\nabla_\beta L$.

Write $x^{(k)}$ for token k as a column, so row k of X is $x^{(k)\top}$, with its own μ_k, v_k, σ_k and normalized $\hat{x}^{(k)}$. $\hat{X}$ has rows $\hat{x}^{(k)\top}$, Y has rows $(\gamma \odot \hat{x}^{(k)} + \beta)^\top$, G has rows $g^{(k)\top}$, and $g'^{(k)} = \gamma \odot g^{(k)}$.

Row k of Y depends on X only through row k of X.	Each row's μ_k and σ_k are computed from that row alone, and γ and β are not functions of X.
Row k of $\nabla_X L$ is $\frac{1}{\sigma_k}\big(g'^{(k)} - \operatorname{mean}(g'^{(k)})\mathbf{1} - \hat{x}^{(k)} \operatorname{mean}(g'^{(k)} \odot \hat{x}^{(k)})\big)^\top$.	By step 1 the only path from row k of X to L is through row k of Y, whose upstream gradient is $g^{(k)}$; the rest is Problem 7 for token k.
$\partial L/\partial \gamma_j = \sum_k G_{kj}\hat{X}_{kj}$ and $\partial L/\partial \beta_j = \sum_k G_{kj}$.	γ_j and β_j appear in Y_{kj} for every row k, with coefficients $\hat{X}_{kj}$ and 1; the chain rule sums over every entry of Y that contains them.
$\big((G \odot \hat{X})^\top \mathbf{1}\big)_j = \sum_k (G \odot \hat{X})_{kj}$ and $(G^\top \mathbf{1})_j = \sum_k G_{kj}$, with $\mathbf{1} \in \mathbb{R}^N$.	For a matrix M with N rows, $M^\top \mathbf{1}$ adds up the N rows of M; shapes $(n \times N)(N \times 1) = n \times 1$.

$\nabla_X L$ is Problem 7 applied to each row; $\nabla_\gamma L = (G \odot \hat{X})^\top \mathbf{1}$; $\nabla_\beta L = G^\top \mathbf{1}$	$N \times n$, n and n, the shapes of X, γ and β. The sums over the N rows appear because γ and β are shared by every token; $\nabla_X L$ has no sum, because each token has its own row of X. In array code, the two means in step 2 are taken along the feature axis, one per row.

Problem 10

> RMSNorm drops the mean: $y = \gamma \odot x/r$ with $r = \sqrt{\frac{1}{n}\sum_i x_i^2 + \epsilon}$. Compute $\nabla_x L$.

Here write $\hat{x} = x/r$ for the RMS-normalized input, and keep $g' = \gamma \odot g$.

$\nabla_{\hat{x}} L = g'.$	$y = \gamma \odot \hat{x}$, so Problem 5 applies unchanged; there is no β, and it would not matter if there were.
$\nabla_x r = \dfrac{1}{2r} \cdot \dfrac{2}{n}\, x = \dfrac{1}{n}\hat{x}.$	$r = \sqrt{\frac{1}{n}\sum_i x_i^2 + \epsilon}$, the derivative of $\frac{1}{n}\sum_i x_i^2$ with respect to x_j is $\frac{2}{n}x_j$, and $x/r = \hat{x}$. Problem 6, step 2, with x in place of c.
$\partial\hat{x}/\partial x = \frac{1}{r} I - \frac{1}{r^2}\, x\, (\nabla_x r)^\top = \frac{1}{r}\big(I - \frac{1}{n}\hat{x}\hat{x}^\top\big).$	Problem 6, step 3, with $\partial x/\partial x = I$ in place of $\partial c/\partial x = P$: nothing is subtracted, so there is no centering term.
$\nabla_x L = \frac{1}{r}\big(g' - \frac{1}{n}\hat{x}(\hat{x}^\top g')\big).$	The Jacobian is symmetric, so apply it to g' as in Problem 7.
$\nabla_x L = \frac{1}{r}\big(g' - \hat{x}\,\mathrm{mean}(g' \odot \hat{x})\big)$ with $\hat{x} = x/r$	$n \times 1$. It is Problem 7 without the $\mathrm{mean}(g')\mathbf{1}$ term. Because RMSNorm does not subtract the mean, a shift of x does change its output, and $\mathbf{1}^\top \nabla_x L$ is not zero in general; only the rescale direction is (nearly) invisible, as in Problem 8.

Where this goes wrong

1. Treating μ and σ as constants

Per feature, layer norm looks like an affine map, $x_i \mapsto \gamma_i(x_i - \mu)/\sigma + \beta_i$, and an affine map's input gradient is just its slope times the upstream gradient.

$\nabla_{\hat{x}} L = g'$	Right so far: Problem 5.
"$\hat{x} = (x - \mu\mathbf{1})/\sigma$ subtracts a number and divides by a number, so $\partial\hat{x}/\partial x = I/\sigma$."	The shortcut that causes the mistake: μ and σ are treated as numbers fixed in the forward pass, when both are functions of every x_i.
$\nabla_x L = g'/\sigma$ ✗	That is the Jacobian's first term only (Problem 6); the other two come from $\nabla_x \mu$ and $\nabla_x \sigma$, and the centering term is what makes $\mathbf{1}^\top \nabla_x L = 0$ (Problem 8). This answer sums to $\mathbf{1}^\top g'/\sigma$, so it claims that a uniform shift of x changes the loss, which it cannot.

2. Gradient of γ from the output instead of the normalized input

In a linear layer a weight's gradient is the upstream gradient times the layer's input, and it is easy to reach for the nearest saved vector instead.

$y_i = \gamma_i \hat{x}_i + \beta_i$ and $\partial L/\partial y_i = g_i$	Right so far: the forward pass and the upstream gradient.
"The gradient of a scale is the upstream gradient times what it scales, and what comes out is y."	The shortcut that causes the mistake: using the saved output of the layer in place of the input that γ multiplies.
$\nabla_\gamma L = g \odot y$ ✗	y already contains γ: $g \odot y = \gamma \odot g \odot \hat{x} + g \odot \beta$. The coefficient of γ_i in y_i is $\hat{x}_i$, so the answer is $g \odot \hat{x}$ (Problem 4). The two agree at the usual initialization, $\gamma = \mathbf{1}$ and $\beta = 0$, so the bug passes a gradient check at step 0.

3. Dividing only the first term by σ

Written out term by term, Problem 7's leading $\frac{1}{\sigma}$ is easy to attach to the first term only.

$\partial\hat{x}/\partial x = \frac{1}{\sigma}\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\hat{x}\hat{x}^\top\right)$	Right so far: Problem 6.
"Apply it to g': g'/σ, then subtract the mean of g' and the $\hat{x}$ term."	The slip that causes the mistake: scaling only the identity term, as if $\frac{1}{\sigma}$ belonged to I.
$\nabla_x L = g'/\sigma - \mathrm{mean}(g')\mathbf{1} - \hat{x}\,\mathrm{mean}(g' \odot \hat{x})$ ✗	The $\frac{1}{\sigma}$ multiplies the whole Jacobian, so every term. This answer sums to $(\frac{1}{\sigma} - 1)\mathbf{1}^\top g'$, not 0, and is right only when $\sigma = 1$, so it looks right on rows already close to unit variance.

4. Forgetting to sum γ's gradient over the batch

For one token $\nabla_\gamma L = g \odot \hat{x}$ (Problem 4); in a batch g and $\hat{x}$ become matrices.

$\partial L/\partial\gamma_j$ collects a term $G_{kj}\hat{X}_{kj}$ from every row k	Right so far: Problem 9, step 3.
"Replace the vectors by their batched matrices."	The shortcut that causes the mistake: $\nabla_X L$ is the per-token formula row by row, and the same substitution is applied to a parameter.
$\nabla_\gamma L = G \odot \hat{X}$ ✗	γ is shared by every row, so its gradient is the sum of the per-row gradients, $(G \odot \hat{X})^\top\mathbf{1}$ (Problem 9): length n, not $N \times n$. In array code the update $\gamma - \eta\, G \odot \hat{X}$, with learning rate η, broadcasts silently and turns γ into one scale per token.

More at eulerpad.app. This is one of 82 free practice sets with worked solutions at eulerpad.app/practice, alongside guides to the topics behind them.

EulerPad, the app, reads your handwritten math on iPad, iPhone and Android and checks it line by line. It's free to start.

This set is also Chapter 12 of *Deep Learning Math by Hand*, a 250-problem workbook with worked solutions: eulerpad.app/workbooks

eulerpad.app/practice/layer-norm-backward