CSCI 145 Problem 17: Finetuning Through a Random Sketch

July 25, 2026

Submission Instructions

Please upload your work to Gradescope by **11:59pm on DATE**.

- Turn in a single PDF of *handwritten* solutions. Typed (including LaTeX) solutions will not be accepted.
- Problems are grouped into three-week **cycles**. Once per cycle, you will meet with the grader and present the solution to *one* problem from that cycle, selected at random when you arrive — so bring your work and be ready to explain any of them.

Problem 17: Finetuning Through a Random Sketch

LoRA starts training with $\mathbf{B} = \mathbf{0}$, so the adapter contributes nothing and the model begins exactly at the pretrained weights. But if $\mathbf{B} = \mathbf{0}$, then the rk parameters in $\mathbf{A}$ have nothing multiplying them, and lecture claimed their gradient is zero: on the first step, only $\mathbf{B}$ moves at all. A method whose first step can move only one of its two factors ought to be badly handicapped. This problem works out what that first step actually computes, and finds it is the full finetuning gradient viewed through a random rank-r sketch.

Throughout, we use the reading's notation. The frozen pretrained matrix is $\mathbf{W} \in \mathbb{R}^{d\times k}$, the adapter is $\Delta\mathbf{W} = \mathbf{BA}$ with $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$, and the learning rate is $\alpha > 0$. Write

$$\mathbf{G} = \nabla_{\Delta\mathbf{W}}\mathcal{L} \in \mathbb{R}^{d\times k}$$

for the gradient of the loss with respect to the full update $\Delta\mathbf{W}$, treating its dk entries as free: this is exactly the gradient a full finetuning run would follow. Superscripts in parentheses index training steps, as usual, so $\mathbf{A}^{(0)}$ is the random matrix $\mathbf{A}$ is initialized to.

Part A: Only B Moves at First

The loss sees $\mathbf{B}$ and $\mathbf{A}$ only through their product, so differentiate through it. Write $[\Delta\mathbf{W}]_{i,j} = \sum_{m=1}^{r}[\mathbf{B}]_{i,m}[\mathbf{A}]_{m,j}$ and sum over every route from one entry to the loss (the same two moves the backpropagation lecture used to reach $\nabla_{\mathbf{W}}\mathcal{L} = (\nabla_{\mathbf{v}}\mathcal{L})\,\mathbf{u}^\top$), and show that

$$\nabla_{\mathbf{B}}\mathcal{L} = \mathbf{G}\mathbf{A}^\top, \qquad \nabla_{\mathbf{A}}\mathcal{L} = \mathbf{B}^\top\mathbf{G}.$$

Check that the two shapes come out right. Now evaluate both at the standard initialization $\mathbf{B}^{(0)} = \mathbf{0}$ with $\mathbf{A}^{(0)}$ a nonzero random matrix, and conclude that $\nabla_{\mathbf{A}}\mathcal{L} = \mathbf{0}$ while $\nabla_{\mathbf{B}}\mathcal{L}$ is generally nonzero, so the first gradient step moves $\mathbf{B}$ and leaves $\mathbf{A}$ exactly where it started.

Part B: The First Update Is the Gradient Through a Random Sketch

By Part A, after one step of plain gradient descent $\mathbf{B}^{(1)} = \mathbf{0} - \alpha\,\mathbf{G}\mathbf{A}^{(0)\top}$ while $\mathbf{A}^{(1)} = \mathbf{A}^{(0)}$. Compute the resulting update to the effective weight matrix, $\Delta\mathbf{W}^{(1)} = \mathbf{B}^{(1)}\mathbf{A}^{(1)}$, and show

$$\Delta\mathbf{W}^{(1)} = -\alpha\,\mathbf{G}\big(\mathbf{A}^{(0)\top}\mathbf{A}^{(0)}\big),$$

against full finetuning's first step, $\Delta\mathbf{W}^{(1)}_{\text{full}} = -\alpha\,\mathbf{G}$. LoRA therefore computes the very same $\mathbf{G}$ that full finetuning does, and then multiplies it by the fixed matrix $\mathbf{A}^{(0)\top}\mathbf{A}^{(0)} \in \mathbb{R}^{k\times k}$. Show that this matrix has rank at most r and is positive semi-definite, then explain in one or two sentences what it does to $\mathbf{G}$: which part of the full gradient survives the multiplication, and which part is discarded entirely. (Substituting the SVD of $\mathbf{A}^{(0)}$ makes both facts immediate, and tells you what the surviving directions are.)

Part C: Eckart–Young–Mirsky Bounds the Gap

Part B shows LoRA reaches a rank-r update chosen by a *random* sketch. Compare that against the best rank-r update there is. Let $\Delta\mathbf{W}^\star \in \mathbb{R}^{d\times k}$ be the ideal update, the one full finetuning would eventually converge to, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots$. Suppose we could choose $\mathbf{B}$ and $\mathbf{A}$ optimally rather than by gradient descent, so that $\mathbf{BA}$ is the closest rank-r matrix to $\Delta\mathbf{W}^\star$ in Frobenius norm. Using the Low-rank Approximation lecture's Eckart–Young–Mirsky theorem, write

down that optimal $\mathbf{BA}$ and the Frobenius-norm error it leaves behind, in terms of the σ_i. Evaluate that error for a spectrum with $\sigma_i = 1$ for $i \leq r$ and $\sigma_i = 0$ afterwards, and for a flat spectrum with $\sigma_i = 1$ for all $i \leq \min(d, k)$, and say which of the two a rank-r adapter can serve. Then state in one sentence the lesson of Parts A through C: what LoRA is really doing to the finetuning gradient, and what property of $\Delta\mathbf{W}^\star$ decides whether that is good enough.

Where this goes next. Part B's random sketch is the same compression-by-random-projection that appears whenever a large computation has to be pushed through a smaller one, and Part C is this unit's second use of Eckart–Young–Mirsky to bound the error of a low-rank shortcut, after the truncation bound itself. The Structure-informed Optimization lecture asks a related question from the other side: not how to restrict the update to a low-rank correction, but how to take a full-rank step that respects a weight matrix's shape as a linear map rather than a flat list of numbers.