Problem 1: Warmup — Softmax + Matrix Gradients

Part 1: Temperature-scaled softmax Jacobian (6 pts)

Let $z \in \mathbb{R}^n$ and $\tau > 0$. Define

$$y_i = \text{softmax}_\tau(z)_i = \frac{e^{z_i/\tau}}{\sum_{j=1}^n e^{z_j/\tau}}.$$

We want the Jacobian J with $J_{ik} = \frac{\partial y_i}{\partial z_k}$. Write $S = \sum_{j=1}^n e^{z_j/\tau}$ so that $y_i = e^{z_i/\tau}/S$. Then

$$\frac{\partial y_i}{\partial z_k} = \frac{1}{S}\frac{\partial}{\partial z_k}\big(e^{z_i/\tau}\big) \; - \; \frac{e^{z_i/\tau}}{S^2}\frac{\partial S}{\partial z_k}.$$

We have $\frac{\partial}{\partial z_k}(e^{z_i/\tau}) = \frac{1}{\tau}e^{z_i/\tau}\,\mathbf{1}[i=k]$ and $\frac{\partial S}{\partial z_k} = \frac{1}{\tau}e^{z_k/\tau}$. Substituting and using $y_i = e^{z_i/\tau}/S$ and $y_k = e^{z_k/\tau}/S$,

$$\frac{\partial y_i}{\partial z_k} = \frac{1}{\tau}\,y_i\,\mathbf{1}[i=k] \; - \; \frac{1}{\tau}\,y_i\,y_k \; = \; \frac{1}{\tau}\big(y_i\,\mathbf{1}[i=k] - y_i y_k\big).$$

So the (i,k) entry of the Jacobian is $\frac{1}{\tau}(y_i\delta_{ik} - y_i y_k)$, which is exactly the (i,k) entry of $\frac{1}{\tau}\big(\text{diag}(y) - yy^\top\big)$. Hence

$$\boxed{\frac{\partial y}{\partial z} = \frac{1}{\tau}\big(\text{diag}(y) - yy^\top\big).}$$

Part 2: Gradient w.r.t. A and b in affine least-squares (6 pts)

Let $A \in \mathbb{R}^{d\times m}$, $b \in \mathbb{R}^d$, $x \in \mathbb{R}^m$, and $t \in \mathbb{R}^d$ (fixed). Define

$$f(A,b) = \frac{1}{2}\|Ax + b - t\|_2^2.$$

Write the residual as $r = Ax + b - t \in \mathbb{R}^d$. Then $f = \frac{1}{2}r^\top r$, so $\frac{\partial f}{\partial r} = r^\top$, i.e. $\nabla_r f = r$.

f depends on A only through $r = Ax + b - t$. For the (i,j) entry of A, we have $\frac{\partial r}{\partial A_{ij}} = x_j \cdot e_i$ (the ith standard basis vector), because $(Ax)_i = \sum_\ell A_{i\ell}x_\ell$, so $\frac{\partial (Ax)_i}{\partial A_{ij}} = x_j$. By the chain rule,

$$\frac{\partial f}{\partial A_{ij}} = (\nabla_r f)^\top \frac{\partial r}{\partial A_{ij}} = r^\top(x_j e_i) = r_i x_j.$$

So the gradient of f with respect to A has (i,j) entry $r_i x_j$, which is the (i,j) entry of $rx^\top$. Thus

$$\boxed{\nabla_A f = (Ax + b - t)\,x^\top.}$$

f depends on b through $r = Ax + b - t$, and $\frac{\partial r}{\partial b} = I_d$, so $\nabla_b f = \nabla_r f = r$. Hence

$$\boxed{\nabla_b f = Ax + b - t.}$$

Part 3: Softmax vs. sigmoid: when and why? (3 pts)

Softmax, not sigmoid: Multi-class classification with a single label per example (e.g. image classification into one of many classes, or next-token prediction over a vocabulary). The classes are mutually exclusive, so we model a single categorical distribution over classes. Softmax outputs a probability vector that sums to 1 and is the correct parameterization for that distribution; we typically use cross-entropy loss between the one-hot label and the softmax output. Sigmoid would give one probability per class that does not sum to 1 and does not represent a proper categorical distribution, so it is not appropriate here.

Sigmoid, not softmax: Multi-label classification (e.g. tagging documents with multiple topics, or detecting multiple objects in an image). Each label is a binary outcome (present or not), and labels are not mutually exclusive. We model each label independently with a Bernoulli, so the right output is one probability per label via sigmoid. Cross-entropy (or binary cross-entropy) is applied per label. Softmax would force the outputs to sum to 1 and imply mutual exclusivity, which is wrong when several labels can be on at once.

In short: softmax + cross-entropy for a single categorical (mutually exclusive) choice; sigmoid for independent binary labels. The choice of activation and loss follows from whether we assume one-of-K or independent binary labels.