Lecture 4 Part 2: Finite-difference Approximations

MIT 18.S096 Matrix Calculus For Machine Learning and Beyond

March 25, 2024

1 Finite-difference approximation for vector-to-scalar functions

This part largely follows from Section 8.1 of Numerical Optimization by Nocedal and Wright.

There are so many assumptions (colored in red)!

1.1 Truncation error

Consider a twice continuously differentiable function $f: \mathbb{R}^n \to \mathbb{R}$. Let $x, p \in \mathbb{R}^n$. Then, by Taylor's theorem,

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p \quad \text{for some } t \in (0,1).$$

Note that this is actually pretty interesting because $\nabla^2 f(x+tp)$ is the Hessian of a point on the line that interpolates x and p.

Continuing on:

$$\begin{aligned}
f(x+p) &= f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p \\
f(x+p) - f(x) - \nabla f(x)^T p &= \frac{1}{2} p^T \nabla^2 f(x+tp) p \\
\|f(x+p) - f(x) - \nabla f(x)^T p\| &= \left\| \frac{1}{2} p^T \nabla^2 f(x+tp) p \right\| \\
\|f(x+p) - f(x) - \nabla f(x)^T p\| &\leq \frac{1}{2} \|p^T\| \, \|\nabla^2 f(x+tp)\| \, \|p\|.
\end{aligned}$$

Let L be the bound on $\|\nabla^2 f(\cdot)\|$ in the region of interest. Obtain

$$\|f(x+p) - f(x) - \nabla f(x)^T p\| \leq (L/2)\|p\|^2.$$

If we choose p to be εe_i, then $\nabla f(x)^T p = \varepsilon(\partial f / \partial x_i)$ and $\|p\|^2 = \varepsilon^2$. Obtain

$$-(L/2)\varepsilon^2 \leq f(x+\varepsilon e_i) - f(x) - \varepsilon \frac{\partial f}{\partial x_i}(x) \leq (L/2)\varepsilon^2$$

$$-f(x+\varepsilon e_i) + f(x) - (L/2)\varepsilon^2 \leq -\varepsilon \frac{\partial f}{\partial x_i}(x) \leq -f(x+\varepsilon e_i) + f(x) + (L/2)\varepsilon^2$$

$$\frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon} + (L/2)\varepsilon \geq \frac{\partial f}{\partial x_i}(x) \geq \frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon} - (L/2)\varepsilon$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon} + \delta_\varepsilon \quad \text{where} \quad |\delta_\varepsilon| \leq (L/2)\varepsilon.$$

δ_ε is commonly referred to as the *truncation* error. This becomes *forward difference* formula if we ignore the δ_ε term, which becomes smaller and smaller as $\varepsilon \to 0$.

1.2 Round-off error

For simplicity, assume that the relative error in the computed f is bounded by $\boldsymbol{u}$ ($\boldsymbol{u}$ is about 10^{-16} in double-precision representation) (I don't really know when this assumption is reasonable.):

$$\begin{aligned}|\mathrm{comp}(f(x)-f(x))| &\leq \boldsymbol{u}L_f\\ |\mathrm{comp}(f(x+\varepsilon e_i)-f(x+\varepsilon e_i))| &\leq \boldsymbol{u}L_f,\end{aligned}$$

where $\mathrm{comp}(\cdot)$ denotes the computed value, and L_f is a bound on the value of $|f(\cdot)|$ in the region of interest. If we use the computed values in the forward difference formula

$$\frac{\partial f}{\partial x_i}(x)=\frac{f(x+\varepsilon e_i)-f(x)}{\varepsilon}+\delta_\varepsilon,$$

we get

$$|\delta_\varepsilon|\leq \underbrace{(L/2)\varepsilon}_{\text{truncation error}}+\underbrace{2\boldsymbol{u}L_f/\varepsilon}_{\text{round-off error}}.$$

Notice how the truncation error decreases as $\varepsilon\to 0$, while the round-off error blows up as $\varepsilon\to 0$.

Taking the derivaive of this expression with respect to ε and setting it to zero, we obtain

$$\varepsilon^*=\sqrt{\frac{4L_f\boldsymbol{u}}{L}}.$$

Assuming that $4L_f/L\approx 1$, we get

$$\varepsilon^*=\sqrt{\boldsymbol{u}},$$

which is what's used in most packages. In PyTorch, $\boldsymbol{u}=1\times 10^{-6}$.

2 Finite-difference approximation for matrix-to-matrix functions

Let v be a generic vector (could be a scalar, a vector or a matrix). Then, in the differential notation, we have

$$f(v+dv)-f(v)=f'(v)[dv]+\text{higher-order terms}.$$

When dv is very small, we have

$$f(v+dv)-f(v)\approx f'(v)[dv].$$

$f'(v)[\cdot]$ would typically be something that we derive by hand or autograd – we can check the correctness of this derivation by first choosing a small dv and then comparing $f(v+dv)-f(v)$ and $f'(v)[dv]$ – their difference should be small.

How should we measure their difference? Ratio of norms:

$$\frac{\|\text{estimated}-\text{truth}\|}{\|\text{truth}\|}$$

For matrices, we would use the Frobenius norm.

$f(v+dv)-f(v)$ is called the *forward* difference. We also could have chosen $f(v)-f(v-dv)$, the *backward* difference. But it turns out that the *central* difference usually works the best:

$$f\left(v+\frac{1}{2}dv\right)-f\left(v-\frac{1}{2}dv\right).$$