On the choice of inner product for reverse-mode autodiff

Forward mode automatic differentiation transforms a program which computes a function $f : X \to Y$ into a program that returns the primal value $y = f(x)$ along with the directional derivative $\dot{y} = \mathbb{D}f[x](\dot{x})$ in some given direction $\dot{x}$.

In reverse mode, we obtain a program which computes the primal value along with the *adjoint* of the directional derivative operator $\mathbb{D}f[x]^* : Y \to X$. We *then* evaluate this operator to obtain a final derivative $\overline{x} = \mathbb{D}f[x]^*(\overline{y})$ given some $\overline{y}$.

In Mooncake.jl, the operator $\mathbb{D}f[x]^*$ is the `pb!!` closure in `out, pb!! = rule(fx_fwds...)` and $\overline{x}$ is the second return value of `value_and_pullback!!(rule, ȳ, f, x...)`.

We tend to treat $\dot{y}$ (returned by forward-mode) and $\overline{x}$ (returned by reverse-mode) as the same. We should be careful, because they do not strictly belong to the same space. Instead, there is one more step we should do to recover $\dot{y}$ from $\overline{x}$ after reverse-mode. We tend to skip this step because, with the standard adjoint operator, $\dot{y}$ and $\overline{x}$ both look the same.

Adjoints and inner products

Reminder. If V is a vector space, then its *dual space* V^* is the vector space of linear operators from V to the underlying field.

The adjoint $\mathbb{D}f[x]^*$ is dependent on a choice of inner products on the vector spaces X and Y. This choice is usually implicit, even though from the defining relation of the adjoint

$$\langle \mathbb{D}f[x]^*(\overline{y}), \dot{x} \rangle_X = \langle \overline{y}, \mathbb{D}f[x](\dot{x}) \rangle_Y \tag{1}$$

it is clear that different choices of inner product result in different operators $\mathbb{D}f[x]^*$.

Note on notation

We inherit notation from Mooncake.jl. This includes “dot and bar” notation (prevalent in the autodiff community):

- The “dot” tangent vector $\dot{x}$ behaves like $\mathrm{d}x$, so that $\dot{y} = \frac{\partial y}{\partial x}\dot{x}$;
- The “bar” tangent vector $\overline{x}$ behaves like $\frac{\partial}{\partial x}$, so that $\overline{y} = \frac{\partial x}{\partial y}\overline{x}$.

If $\dot{x}$ is represented as a column vector, then $\overline{x}$ is naturally represented as a row vector.

The tangent and cotangent spaces are isomorphic, but a choice of isomorphism is a choice of adjoint is a choice of inner product (For example, the transpose $x \mapsto x^T$ defines the Euclidean inner product $\langle x, y \rangle = x^T y$. Another choice is the map $(t, x, y, z) \mapsto (-t, x, y, z)^T$ and the Lorentian inner product.)

Does the choice of inner product matter?

Suppose $y = f(x)$. The reverse-pass yeilds $\overline{x} = \mathbb{D}f[x]^*(\overline{y})$ for an initial $\overline{y}$. We are interested in the directional derivatives $\dot{y} = \mathbb{D}f[x]^*(\dot{x})$ for each linearly independent $\dot{x}$. Using Equation 1, we can obtain $\dot{y}$ as

$$\langle \overline{x}, \dot{x} \rangle = \langle \overline{y}, \dot{y} \rangle \tag{2}$$

At first glance it is not obvious that reverse-mode differentiation is independent of the inner products involved.

We care about the actual derivative $\mathbb{D}f[x]$, not the adjoint $\mathbb{D}f[x]^*$. What we really do in reverse mode is use Equation 1 to recover the derivative $\mathbb{D}f[x](\dot{x})$ in terms of $\mathbb{D}f[x]^*(\overline{y})$ by fixing various values of $\overline{y}$ and $\dot{x}$.

Indeed, the original choice of inner product is arbitrary. Varying the inner product varies $\mathbb{D}f[x]^*$ — but the inner product must be used again to obtain $\mathbb{D}f[x](\dot{x})$, and this ‘cancels out’ the dependence on the inner product.

Examples to illustrate

When $f : \mathbb{R}^N \to \mathbb{R}$, we chose $\overline{y} = 1$ to obtain

$$\langle \mathbb{D}f[x]^*(1), \dot{x} \rangle = \mathbb{D}f[x](\dot{x}). \tag{3}$$

After computing $\overline{x} \coloneqq \mathbb{D}f[x]^*(1)$ with a single reverse pass, we simply evaluate Equation 3 for each standard basis vector $\dot{x} \in \{\dot{e}_1, ..., \dot{e}_N\}$ in order to obtain the full gradient

$$\nabla f[x] \coloneqq \begin{pmatrix} \mathbb{D}f[x](\dot{e}_1) \\ \vdots \\ \mathbb{D}f[x](\dot{e}_N) \end{pmatrix} = \begin{pmatrix} \langle \overline{x}, \dot{e}_1 \rangle \\ \vdots \\ \langle \overline{x}, \dot{e}_N \rangle \end{pmatrix}. \tag{4}$$

When $f : \mathbb{R}^N \to \mathbb{R}^M$, we compute $\mathbb{D}f[x](\overline{e}_i)$ once for each standard basis vector $\overline{e}_i$ of $\mathbb{R}^M$. Then, instead of Equation 3, we have

$$\begin{pmatrix} \langle \mathbb{D}f[x]^*(\overline{e}_1), \dot{x} \rangle \\ \vdots \\ \langle \mathbb{D}f[x]^*(\overline{e}_M), \dot{x} \rangle \end{pmatrix} = \mathbb{D}f[x](\dot{x}) \in \mathbb{R}^M \tag{5}$$

which we may then evaluate for each $\dot{x} \in \{\dot{e}_1, ..., \dot{e}_N\}$ to recover the “gradient” (which is now an N-vector of M-vectors).