2/13/2023 (1)

Matrix Calculus Part II

Linear Transformation View

$f(x + dx) - f(x) \approx f'(x)[dx]$

Numerically perhaps take $dx = 1e-7$ or just .0000001

Abuse of notation:

1. Scalar → Scalar

$df = \underset{\text{lin transform}}{f'(x)[dx]} = \underset{\text{scalar}}{f'(x)} \cdot dx$

e.g. $f(x) = \sin x$ $\{\sin x\}'[dx] = (\cos x)\, dx$

2. Vector → scalar

$df = f'(x)[dx] = \underbrace{f'(x)}_{\text{row vector}}\, dx$ $f'(x) = (\nabla_x f)^T$

e.g. $f(x) = x^T x$

$\underset{\text{lin transform}}{f'(x)[dx]}$ $\underbrace{2x^T}_{\text{row vector}}\, dx$

3. vector → vector

$df = f'(x)[dx] = \underbrace{J}_{\text{Jacobian matrix}}\, dx$ $(= f'(x)\, dx)$ $\left(J \equiv f_x \equiv \frac{\partial f}{\partial x}\right)$

4 matrix → Scalar

$df = f'(x)[dX] = \operatorname{tr} G^T dX$ $G = \nabla_x f$

e.g. $f(X) = \operatorname{tr}(X)$ $G = I$

5 matrix → matrix

e.g. $d(X^2)$: $(X^2)'[dX] = X\,dX + dX\,X$

Rules

plus/Minus $\quad f(x) = g(x) \pm h(x)$

$$f'[dx] = g'[dx] \pm h'[dx]$$

or $\quad f' = g' \pm h' \quad$ or $\quad f'(x)[dx] = g'(x)[dx] \pm h'(x)[dx]$

Product (no matter what kind) $\quad f(x) = g(x)\, h(x)$

$$f'[dx] = g'[dx]\, h(x) + g(x)\, h'[dx]$$

& the most important chain rule

$$f(x) = g(h(x))$$

$$f'(x) = g'(h(x)) \underset{\substack{\uparrow \\ \text{composition}}}{\circ} h'(x)$$

or $\quad f'(x)[dx] = g'(h(x))\left[h'(x)[dx]\right]$

Try to avoid indices: the "grown up" approach to matrix calculus

$x \in \mathbb{R}^n$: $\quad d(x^T x) = x^T dx + dx^T x = 2x^T dx$

dot products of vectors commute

$X \in \mathbb{R}^{n,n}$ $\quad d(X^2) = X\,dX + dX\,X$

matrices don't commute

2/13/2023 (2)

Vector to Vector Examples

$x \in \mathbb{R}^2$

1. "rotate by θ" $\quad f_1(x) = \overbrace{\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}}^{R(\theta)} \cdot x$

2. warp(θ) $\quad f_2(x) = R(\theta \|x\|)\, x$

the further from the origin, the more you twist

1 $\quad f_1(x) = R(\theta)\, x$

$d f_1(x) = R(\theta)\, dx$

$f_1' = R(\theta)$

2 $\quad d\left(R(\theta \|x\|)\, x \right) = \sim\sim\, x + R(\theta \|x\|)\, dx$

$$d[R(\theta)] = \overbrace{\begin{pmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix}}^{R'(\theta)} d\theta$$

$$d\left((x^T x)^{1/2} \right) = \frac{1}{2} (x^T x)^{-1/2}\, d(x^T x)$$

$$= \frac{x^T dx}{\|x\|}$$

$$d(\text{warp}(\theta) \cdot x) = \theta\, R'(\theta \|x\|)\, x x^T dx + R(\theta)\, dx$$

$$= \left[\theta\, R'(\theta \|x\|)\, x x^T + R(\theta) \right] dx$$

Chain Rule for vectors

$$\begin{array}{l} x \in \mathbb{R}^n \\ \downarrow f \\ y \in \mathbb{R}^K \\ \downarrow g \\ z \in \mathbb{R}^m \\ \downarrow h \\ l \in \mathbb{R} \end{array}$$

$$l = h(g(f(x)))$$

$$dl = \underset{1 \times m}{J_h(z)} \; \underset{m \times K}{J_g(y)} \; \underset{K \times n}{J_f(x)} \, dx$$

$$J_h = \nabla_z h^T$$

$$\nabla_x l = \nabla_x (h \circ g \circ f) = \underset{\text{mtx}}{J_f^T} \; \underset{\text{mtx}}{J_g^T} \; \underset{\text{col}}{J_h^T}$$

Which way is best to form the m numbers of $\nabla_x l$?

$J_h(J_g J_f)$ forward mode $\quad 2\left[(mKn) + (m^2 n)\right]$

$(J_h J_g) J_f$ reverse mode $\quad 2\left[mK + mn\right]$

Many inputs one out $\to$ reverse mode wins
(typical for ML

one input many out $\to$ forward mode wins

Notation

JVP - Jacobian vector product $\quad f'(x) \cdot v$

VJP - vector Jacobian product $\quad v^T f'(x)$

③

Matrix Linear Operators

$\cancel{A}\ X \in \mathbb{R}^{n_1, n_2} \qquad Y \in \mathbb{R}^{m_1, m_2}$

flattening: Can always write $\mathrm{vec}(dY) = \underset{m_1 m_2 \times n_1 n_2}{\boxed{}}\ \mathrm{vec}(dx)$

I don't love flattening
but it is kind of a "least common denominator"

Kronecker Product Notation

In Julia: $\quad \mathrm{kron}(A, B) * \mathrm{vec}(X) \equiv \mathrm{vec}(BXA^T)$

I like $\quad (A \otimes B)[X] = BXA^T$
without writing the vec's

thus
$$X\,dX = (I \otimes X)[dX]$$
$$dX\,X = (X^T \otimes I)[dX]$$

so
$$\underset{\text{operator}}{(X^2)'} = \underset{\text{operator}}{\underline{I \otimes X + X^T \otimes I}}$$

$$X^{-1}X = I$$
$$d(X^{-1})\,X + X^{-1}\,dX = 0$$
$$d(X^{-1}) = -X^{-1}\,dX\,X^{-1}$$
$$(X^{-1})' = -X^{-T} \otimes X^{-1}$$

2/13/2023

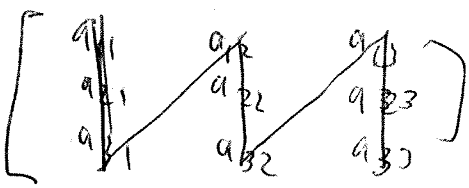
(4)

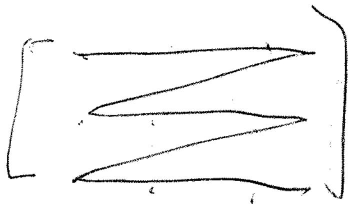
Part II: Optimizing Serial Code

a) Column Major vs Row Major

Column Major: Julia, Matlab, Fortran

Row Major: Python, C





$[a_{11}\ a_{21}\ a_{31}\quad a_{12}\ a_{22}\ a_{32}\quad a_{13}\ a_{23}\ a_{33}]$ vs $[a_{11}\ a_{12}\ a_{13}\quad a_{21}\ a_{22}\ a_{23}\quad a_{31}\ a_{32}\ a_{33}]$

* Linear Algebra Libraries
Arrays are linear in Memory

Compare

```
    for i=1:n, j=1:n
vs  for j=1:n, i=1:n
        c_ij = a_ij + b_ij
    end
```

Multicore Memory Architecture
L1, L2 cache

cache miss – when data needs to be pulled from main memory

Cache aware algorithm - indexing structure chosen by programmer explicitly to ~~that~~ avoid cache misses

cache oblivious algorithm - indexing structure misses cache by design but implicitly

Main Memory

Stack — Data with known sizes "Statically sized" — FAST

Requirement: size must be known at compile time

Heap — Pointers to memory of sizes unknown at compile time "Dynamically sized" — SLOW

Mutation: use memory already preallocated

By convention denoted with a "!" in julia

Broadcasting

Using Static Arrays

~~@s~~ @SVector

@view