Reverse-Mode 自動微分を理解する

Hiromi ISHII

2023-03-15

Tsukuba Computer Mathematics Seminar 2023

Forward-Mode と Reverse-Mode 自動微分

自動微分とは？

- **自動微分**：与えられた関数の値とその微分係数を，合成関数の微分公式（**連鎖律**）を使って効率的・厳密に計算する方法
  - 関数の値を順に計算しながら、連鎖律を使って微分係数も計算する
- 関連するが異なる計算方法：
  - **記号微分**：数式を記号的な構文木で表現し，微分法則に従い記号的に処理.
    - 厳密だが項数が組合せ爆発して非効率になりがち.
  - **数値微分**：極限 $\lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$ の Δx を微小数で置き換えて近似.
    - あくまで近似. Δt をどう近似するかで振る舞いが変わってくる.

連鎖律

- 皆さんお馴染の基本公式

$$\mathbb{R}^n \xrightarrow{f} \mathbb{R}^k \xrightarrow{g} \mathbb{R}^m$$

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}}(g \circ f) = \frac{\mathrm{d}g}{\mathrm{d}f}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$$

- 統一性のため $\boldsymbol{u}_0 = \boldsymbol{x}, \boldsymbol{u}_1 = f(\boldsymbol{u}_0), \boldsymbol{u}_2 = g(\boldsymbol{u}_1) = g(f(\boldsymbol{u}_0))$ として並び換えれば,

$$\mathbb{R}^m \xleftarrow{\boldsymbol{u}_2} \mathbb{R}^k \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^n \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\mathrm{d}\boldsymbol{u}_2}{\mathrm{d}\boldsymbol{u}_0} = \frac{\mathrm{d}\boldsymbol{u}_2}{\mathrm{d}\boldsymbol{u}_1}\frac{\mathrm{d}\boldsymbol{u}_1}{\mathrm{d}\boldsymbol{u}_0}$$

連鎖律

- 皆さんお馴染の基本公式

$$\mathbb{R}^n \xrightarrow{\ f\ } \mathbb{R}^k \xrightarrow{\ g\ } \mathbb{R}^m$$

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}}(g \circ f) = \frac{\mathrm{d}g}{\mathrm{d}f}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$$

- 統一性のため $\boldsymbol{u}_0 = \boldsymbol{x}, \boldsymbol{u}_1 = f(\boldsymbol{u}_0), \boldsymbol{u}_2 = g(\boldsymbol{u}_1) = g(f(\boldsymbol{u}_0))$ として並び換えれば,

$$\mathbb{R}^m \xleftarrow{\ \boldsymbol{u}_2\ } \mathbb{R}^k \xleftarrow{\ \boldsymbol{u}_1\ } \mathbb{R}^n \xleftarrow{\ \boldsymbol{u}_0\ } \mathbb{R}^0$$

$$\frac{\mathrm{d}\boldsymbol{u}_2}{\mathrm{d}\boldsymbol{u}_0} = \frac{\mathrm{d}\boldsymbol{u}_2}{\mathrm{d}\boldsymbol{u}_1}\frac{\mathrm{d}\boldsymbol{u}_1}{\mathrm{d}\boldsymbol{u}_0}$$

連鎖律　（多変数）

多変数も同様（但し、各 $\partial \boldsymbol{u}_i / \partial \boldsymbol{u}_j$ はヤコビ行列だと思うこと）

$$\mathbb{R}^{n_N} \xleftarrow{\boldsymbol{u}_N} \mathbb{R}^{n_{N-1}} \xleftarrow{\boldsymbol{u}_{N-1}} \dots \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^{n_0} \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_{N-1}} \frac{\partial \boldsymbol{u}_{N-1}}{\partial \boldsymbol{u}_{N-2}} \dots \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{u}_1} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{u}_0}$$

連鎖律　（多変数）

多変数も同様（但し、各 $\partial \boldsymbol{u}_i / \partial \boldsymbol{u}_j$ はヤコビ行列だと思うこと）

$$\mathbb{R}^{n_N} \xleftarrow{\boldsymbol{u}_N} \mathbb{R}^{n_{N-1}} \xleftarrow{\boldsymbol{u}_{N-1}} \dots \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^{n_0} \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_N} \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_{N-1}} \frac{\partial \boldsymbol{u}_{N-1}}{\partial \boldsymbol{u}_{N-2}} \cdots \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{u}_1} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{u}_0}$$

連鎖律　（多変数）

多変数も同様（但し、各 $\partial \boldsymbol{u}_i / \partial \boldsymbol{u}_j$ はヤコビ行列だと思うこと）

$$\mathbb{R}^{n_N} \xleftarrow{\boldsymbol{u}_N} \mathbb{R}^{n_{N-1}} \xleftarrow{\boldsymbol{u}_{N-1}} \cdots \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^{n_0} \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_N} \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_{N-1}} \frac{\partial \boldsymbol{u}_{N-1}}{\partial \boldsymbol{u}_{N-2}} \cdots \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{u}_1} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{u}_0}$$

$\longleftarrow$ Forward

これを前から計算するのが Forward-Mode 自動微分

連鎖律　（多変数）

多変数も同様（但し、各 $\partial \boldsymbol{u}_i / \partial \boldsymbol{u}_j$ はヤコビ行列だと思うこと）

$$\mathbb{R}^{n_N} \xleftarrow{\boldsymbol{u}_N} \mathbb{R}^{n_{N-1}} \xleftarrow{\boldsymbol{u}_{N-1}} \dots \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^{n_0} \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_N} \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_{N-1}} \frac{\partial \boldsymbol{u}_{N-1}}{\partial \boldsymbol{u}_{N-2}} \cdots \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{u}_1} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{u}_0}$$

$\longleftarrow$ Forward

$\longrightarrow$ Reverse

これを前から計算するのが Forward-Mode 自動微分

これを後から計算するのが Reverse-Mode 自動微分

Foward Mode 自動微分

- Dual Number $\mathbb{R}[d]/(d^2)$ を使って計算できる
  - $x + yd$ の y が一次微分係数を覚える役割を果す（$f(x) + f'(x)d$ と思う）
- 代数演算はそのまま延長、（区分的に）滑らかな関数についても Weil 環の標準的な C^∞- 環構造が入る
  - 難しく聞こえるが、以下のような感じ：

$$
\begin{aligned}
\sin(x + yd) &= \sin(x) + y\cos(x)d, \\
\cos(x + yd) &= \cos(x) - y\sin(x)d, \\
e^{(x+yd)} &= e^x + ye^x d
\end{aligned}
$$

Forward Mode 計算例

- 次で定まる関数 $f : \mathbb{R} \to \mathbb{R}^3$ の x に関する微分を求めてみよう：

$$f(x) = (\sin(x^2+1), e^{x^2}, x^3 e^{-x})$$

- 入口は $x + (\mathrm{d}x/\mathrm{d}x)d = x + d$ を入れて $f(x+d)$ を評価してみる：

$$\begin{aligned}
&f(x+d) \\
&= (\sin((x+d)^2+1), e^{(x+d)^2}, (x+d)^3 e^{-(x+d)}) \\
&= (\sin(x^2+2xd+1), e^{x^2+2xd}, (x^3+3x^2d)(e^{-x}-e^{-x}d)) \\
&= (\sin(x^2+1)+2x\cos(x^2+1)d, e^{x^2}+2xe^{x^2}d, (x^3+3x^2d)(e^{-x}-e^{-x}d)) \\
&= (\sin(x^2+1), e^{x^2}, x^3e^{x-x}) + \underline{(2x\cos(x^2+1), 2xe^{x^2}, -x^3e^{-x}+3x^2e^{-x})}d
\end{aligned}$$

Forward-Mode だけでいいのでは？

- 入力が一変数、出力が多変数の場合は Forward-Mode で十分効率的
- しかし、機械学習のように $f : \mathbb{R}^N \to \mathbb{R}$ で $N \gg 0$ の場合、Forward-Mode では効率がわるすぎる！
  (a) 素朴にやると、$\mathbb{R}[d]^N$ を用意して、$i < N$ ごとに $(0, \ldots, 0, x + d_i, 0, \ldots)$ みたいなのを何回も渡すことになる
  (b) これを組にして $(\mathbb{R}[d]^N)^N$ について一挙に計算すると one-pass
  (c) しかしいずれにせよ変数の数について自乗のオーダーがかかってしまい非効率！
- $\mathrm{d}\boldsymbol{x}/\mathrm{d}\boldsymbol{x} = \mathrm{d}\boldsymbol{u}_0/\mathrm{d}\boldsymbol{u}_0$ ではなく $\mathrm{d}\boldsymbol{u}_N/\mathrm{d}\boldsymbol{u}_N$ の側から辿ったらどうか？
  - これが Reverse-Mode 自動微分！

Reverse-Mode 自動微分いろいろ

Reverse-Mode 自動微分

- Reverse-Mode の概念図はこうだった

$$\mathbb{R}^{n_N} \xleftarrow{\boldsymbol{u}_N} \mathbb{R}^{n_{N-1}} \xleftarrow{\boldsymbol{u}_{N-1}} \cdots \xleftarrow{\boldsymbol{u}_1} \mathbb{R}^{n_0} \xleftarrow{\boldsymbol{u}_0} \mathbb{R}^0$$

$$\frac{\mathrm{d}\boldsymbol{u}_N}{\mathrm{d}\boldsymbol{u}_0} = \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_N} \frac{\partial \boldsymbol{u}_N}{\partial \boldsymbol{u}_{N-1}} \frac{\partial \boldsymbol{u}_{N-1}}{\partial \boldsymbol{u}_{N-2}} \cdots \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{u}_1} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{u}_0}$$

$\longrightarrow$ Reverse

- 問題：計算の入出力と逆順に辿る必要がある！
  - 微分係数の計算には関数の値そのものが必要、往復する必要がある
    - 機械学習で誤差逆伝播法 (back propagation) と呼ばれるもの
  - これを実現するためにいくつかやり方が知られている