

Metody różniczkowania

Filip Kamiński

30.11.2021

Klasyczna definicja różniczkowania

Definition (Pochodna)

$$f'(x) = \frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

- Sens jedynie dla funkcji $f : \mathbb{R} \rightarrow \mathbb{R}^n$ (przy założeniu, że w przypadku wektorowym dzielenie przebiega skalarnie),
- Różne postacie reguły łańcuchowej,
- Co z przypadkiem $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ lub $f : \mathbb{R}^m \rightarrow \mathbb{R}$?
- Wystarczy do zdefiniowania prostych metod różniczkowania numerycznego.

Numeryczne metody różniczkowania

Najprostszy sposób

Formuły dwupunktowe

Rozważamy $f \in C^2[a, b]$, $x_1 = x_0 + h$, $x_0, x_1 \in [a, b]$, $h \neq 0$.

Z rozwinięcia w szereg Taylora z resztą Lagrange'a otrzymujemy

$$\exists \xi \in [x_0, x_1] : f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2}f''(\xi). \quad (1)$$

Jeżeli $h > 0$ to jest to iloraz progresywny (w przód, ang. *forward-difference formula*).

Jeżeli $h < 0$ to jest to iloraz regresywny (wstecz, ang. *backward-difference formula*).

Na podstawie J. Miller *Ilorazy Różnicowe z Wybrane zagadnienia numeryczne — analiza i dobór algorytmów* pod redakcją Jarosława Wąsa.

Bardziej złożone metody

Formuły trzypunktowe (rzędu $p = 2$)

$$f'(x_0) = \frac{1}{2h} [-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3} f^{(3)}(\xi_0) \quad (2)$$

$$f'(x_0) = \frac{1}{2h} [-f(x_0 - h) + f(x_0 + h)] - \frac{h^2}{6} f^{(3)}(\xi_1) \quad (3)$$

$$f'(x_0) = \frac{1}{2h} [f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)] + \frac{h^2}{3} f^{(3)}(\xi_2). \quad (4)$$

Powyższe wzory ukazują dwie przewagi formuły centralnej (3) nad formułami progresywną (2) i regresywną (4). Koszt obliczeniowy mierzony liczbą obliczanych wartości funkcji f jest o $1/3$ mniejszy, a główna część błędu metody jest dwa razy mniejsza.

Bardziej złożone metody II

Przykłady formuł pięciopunktowych (rzęd $p = 4$)

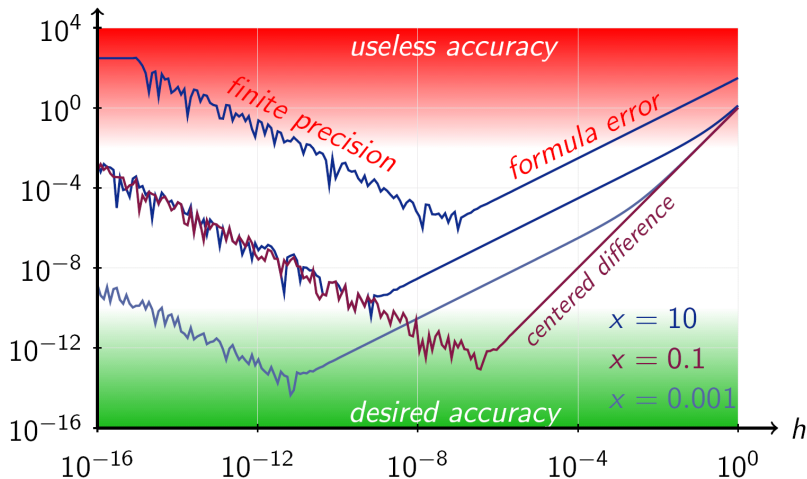
centralna

$$f'(x_0) = \frac{1}{12h} [f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)] + \frac{h^4}{30} f^{(5)}(\xi) \quad (5)$$

progresywna

$$f'(x_0) = \frac{1}{12h} [-25f(x_0) + 48f(x_0 + h) - 36f(x_0 + 2h) + 16f(x_0 + 3h) - 3f(x_0 + 4h)] + \frac{h^4}{5} f^{(5)}(\xi) \quad (6)$$

Błąd różniczkowania



Rysunek: en.wikipedia.org/wiki/Numerical_differentiation

Iloraz z krokiem zespolonym

$$f'(x_0) = \frac{\operatorname{Im}(f(x_0 + ih))}{h} + \frac{h^2}{6} f'''(x_0).$$

Metody symboliczne

Różniczkowanie symboliczne

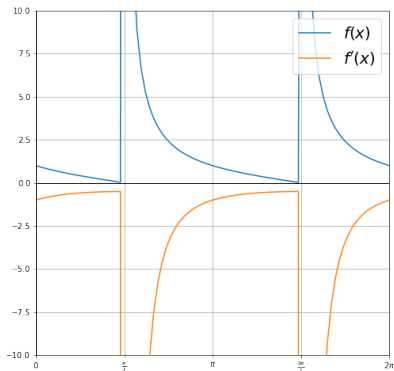
- *MatlabSymbolic Math Toolbox*,
- *Wolfram Alpha/Mathematica*,
- *Maxima*,
- *SymPy*,
- *Maple*.

Rozważmy dość prostą funkcję jednej zmiennej:

$$f(x) = \sqrt{\operatorname{tg}^2(x) + 1} - \operatorname{tg}(x)$$

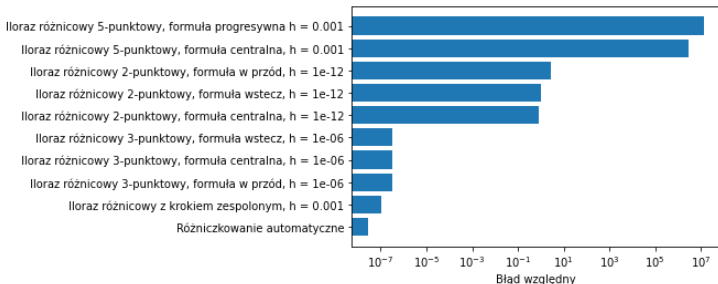
Jej wykres wraz z pochodną pokazano na rysunku 2. Pochodną $f'(x)$ można obliczyć symbolicznie ($\operatorname{tg}(x)' = 1 + \operatorname{tg}^2(x)$):

$$f'(x) = \operatorname{tg}(x)\sqrt{\operatorname{tg}^2(x) + 1} - (\operatorname{tg}^2(x) + 1)$$

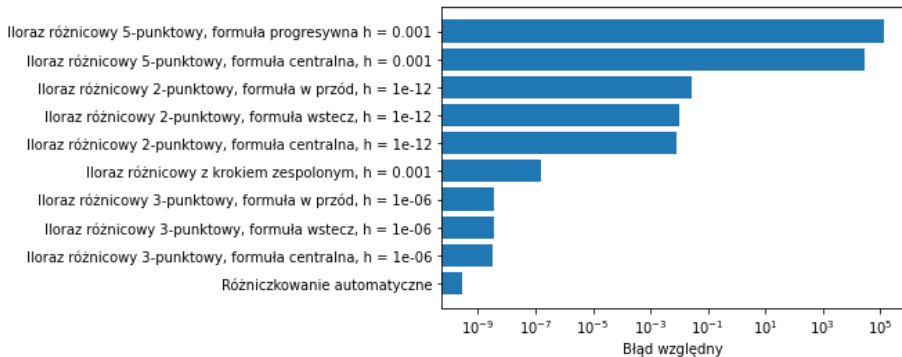


Rysunek: Funkcja $f(x) = \sqrt{\operatorname{tg}(x)^2 + 1} - \operatorname{tg}(x)$ i jej pierwsza pochodna.

W przypadku numerycznego przybliżania wartości pochodnej w okolicach punktów $\frac{\pi}{2} + k\pi, k \in \mathbb{Z}$ mogą pojawić się błędy numeryczne wynikające z odejmowania bliskich wartości.



Rysunek: Błąd względny przybliżenia pochodnej $f'(\frac{\pi}{2} - 10^{-4})$.



Rysunek: Średni błąd względny przybliżenia pochodnej $f'(x)$ w przedziale $x \in [0, \frac{\pi}{2} - 10^{-4}]$ (100 punktów).

Różniczkowanie Automatyczne

Różniczkowanie Automatyczne

Wydajna metoda obliczania pochodnych z dokładnością maszynową (tylko błąd związany z reprezentacją zmiennoprzecinkową).

- Metoda oparta na regule łańcuchowej,
- dobre podstawy matematyczne,
- wymyślona przez R.E. Wengerta w 1964 (podobno pierwsza praca doktorska z informatyki w historii),
- dwa warianty: różniczkowanie w przód (forward mode) i wstecz (reverse mode),
- bardzo wydajna i możliwa do zaimplementowania w prawie każdym środowisku

Kilka definicji

Definition (Graf obliczeniowy)

Grafem obliczeniowym (ang. *computational graph*) nazywamy graf skierowany, w którym wierzchołki zawierają zmienne lub operacje reprezentujące poszczególne elementy badanego wyrażenia. Krawędź pomiędzy dwoma wierzchołkami oznacza związek pomiędzy formułami. Wierzchołek, w którym rozpoczyna się krawędź nazywany jest rodzicem (ang. *parent node*), a połączone z nim wierzchołki nazywane są jego dziećmi (ang. *children nodes*). Dla danej formuły matematycznej może istnieć więcej niż jeden graf obliczeniowy. Poza różniczkowaniem automatycznym grafy obliczeniowe znajdują zastosowanie również w sieciach neuronowych. Czasami, w zależności od sposobu reprezentacji formuły (w większości przypadków formułę można przedstawić w postaci drzewa, a nie grafu), zamiast pojęcia graf obliczeniowy można spotkać się również z określeniem drzewo składniowe (ang. *abstract syntax tree*).

Kilka definicji

Definition (Reguła łańcuchowa)

Podstawą różniczkowania automatycznego jest reguła łańcuchowa. Reguła ta pozwala przedstawić pochodną złożonej funkcji w postaci skończonej kombinacji pochodnych prostszych wyrażeń. Jeżeli

$f : X \rightarrow Y, g : X \rightarrow Z, h : Z \rightarrow Y$ są funkcjami co najmniej pierwszej klasy (C^1) oraz $\dim(X) = \dim(Y) = \dim(Z) = 1$, to reguła łańcuchowa przyjmuje następującą postać:

$$f' = (h \circ g)' = (h' \circ g) \cdot g'$$

co można zapisać w notacji Leibnitza jako:

$$\frac{df}{dx} = \frac{df}{dh} \cdot \frac{dh}{dx}$$

Kilka definicji

Jeżeli $f : X^n \rightarrow Y$, $n \geq 2$, $h_1, \dots, h_n : X \rightarrow Z$, gdzie $f, h_1, \dots, h_n$ są funkcjami co najmniej klasy C^1 i $x_j \in X^n$ to:

$$\frac{\partial f}{\partial x_j} = \frac{\partial}{\partial x_j} f(h_1, \dots, h_n) = \sum_{i=1}^n \frac{\partial f}{\partial h_i} \frac{\partial h_i}{\partial x_j}$$

co można zapisać w postaci wektorowej ($\vec{t} \in X^n$) jako:

$$\frac{d}{dt} f(v(\vec{t})) = \nabla f(v(\vec{t})) \cdot \frac{dv(\vec{t})}{dt}$$

W przypadku funkcji $\mathbb{R}^m \rightarrow \mathbb{R}^n$ ($m, n > 1$) regułę łańcuchową można zapisać za pomocą macierzy Jacobiego:

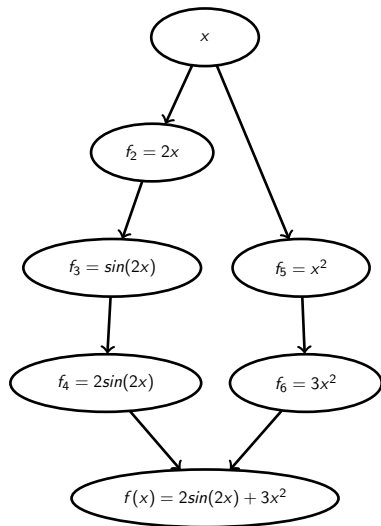
$$J_{f \circ g}(t) = J_f(g(t)) J_g(t)$$

Przykład

Rozważmy przykład

$$f(x) = 2\sin(2x) + 3x^2$$

Metoda w przód

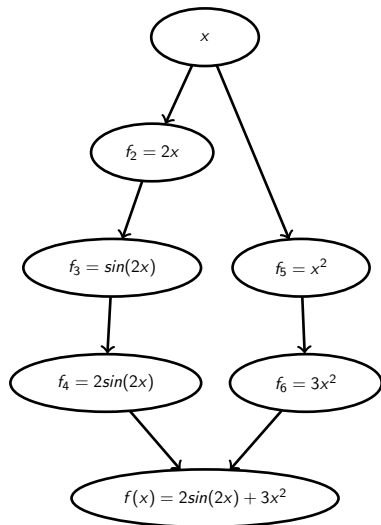


Forward Mode

Rozpoczynamy w x .
Następnie obliczamy
pochodne cząstkowe węzłów
dzieci względem rodziców.

- 1 $\frac{dx}{dx} = 1,$
- 2 $\frac{df_5}{dx} = 2x,$
- 3 $\frac{df_6}{df_5} = 3\frac{df_5}{dx},$
- 4 $\frac{df}{df_6} = \frac{df_6}{df_5},$
- 5 $\dots$
- 6 $4\cos(2x) + 6x$

Metoda wstecz



Metoda wstecz

Rozpoczynamy w $f(x)$ i obliczamy pochodne cząstkowe węzłów względem zmiennej propagując informacje wstecz

- 1 $\frac{df}{dx} = \frac{df}{df_4} \cdot \frac{df_4}{dx} + \frac{df}{df_6} \cdot \frac{df_6}{dx}$,
- 2 $\frac{df_4}{dx} = \frac{df_4}{df_3} \cdot \frac{df_3}{dx}$
- 3 $\frac{df_3}{dx} = \frac{df_3}{df_2} \cdot \frac{df_2}{dx}$,
- 4 $\frac{df_6}{dx} = \frac{df_6}{df_5} \cdot \frac{df_5}{dx}$,
- 5 ...
- 6 $4\cos(2x) + 6x$

Różnice pomiędzy metodą „w przód” i metodą „wstecz” stają się widoczne w przypadku funkcji $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ($n > 1, m > 1$). Znalezienie macierzy Jacobiego funkcji f wymaga n przejść metodą „w przód” lub m przejść metodą „wstecz”.

W przypadku metody „w przód” pojedyncze przejście przez graf obliczeniowy pozwala obliczyć pochodne $\frac{df_j}{dx_i}$ ($j \in \{1, \dots, m\}$) względem i -tej zmiennej ($i \in \{1, \dots, n\}$). Przejście takie odpowiada wypełnieniu i -tej kolumny macierzy Jacobiego (wypenienie całej macierzy wymaga więc n przejść). Metoda ta sprawdza się lepiej w przypadku funkcji, dla których $m \gg n$.

Metoda „wstecz” sprawdza się szczególnie dobrze w przypadku funkcji dla których $n \gg m$. Jedno przejście metodą „wstecz” pozwala na obliczenie pochodnej $\frac{df_j}{dx_i}$ względem wszystkich zmiennych (dla ustalonego j). Pojedyncze przejście metodą „wstecz” odpowiada więc wypełnieniu jednego wiersza w macierzy Jacobiego (tj znalezieniu gradientu funkcji f_j). Znalezienie całej macierzy Jacobiego wymaga wykonania m przejść przez graf. Stosowanie metody „wstecz” wiąże się jednak z koniecznością przechowywania w pamięci większej ilości informacji (większa złożoność pamięciowa). Z tego powodu w praktyce metoda „w przód” okazuje się szybsza w przypadku, gdy m i n są podobne.

Implementacja metody w przód za pomocą liczb dualnych

Jednym ze sposobów na implementację metody różniczkowania „w przód” jest wykorzystanie liczb dualnych (algebry Clifforda). Koncepcja liczb dualnych jest podobna do idei liczb zespolonych. Liczbę dualną z można przedstawić jako parę liczb rzeczywistych (v, v') lub w postaci następującego wyrażenia:

$$z = v + \epsilon v'$$

gdzie

- $\epsilon^2 = 0$ (nilpotent),
- $\epsilon \neq 0$,
- $v, v' \in \mathbb{R}$.

Dodatkowo na liczbach dualnych definiuje się dwie operacje:

- Dodawanie – $(a, a') + (b, b') = (a + b, a' + b')$,
- Mnożenie – $(a, a') * (b, b') = (ab, ab' + a'b)$

Na ich podstawie da się również zdefiniować dzielenie (przy założeniu, że $b \neq 0$):

$$\begin{aligned}\frac{a + \epsilon a'}{b + \epsilon b'} &= \frac{a + \epsilon a'}{b + \epsilon b'} * \frac{b - \epsilon b'}{b - \epsilon b'} = \\ \frac{(a + \epsilon a') * (b - \epsilon b')}{b^2 - \epsilon b' * b + \epsilon b' * b + \epsilon^2 b'^2} &= \frac{ab - \epsilon ab' + \epsilon a'b - \epsilon^2 a'b'}{b^2} = \\ \frac{ab + \epsilon(a'b - ab')}{b^2} &= \frac{a}{b} + \frac{a'b - ab'}{b^2} = \left(\frac{a}{b}, \frac{a'b - ab'}{b^2} \right)\end{aligned}$$

Arytmetyka pierwszych komponentów liczb dualnych przypomina klasyczną arytmetykę. Obliczenia na drugim elementach wydają się odpowiadać regułom różniczkowania. Warto zauważyć, że elementem neutralnym w tej strukturze jest para $(1, 0)$, a dowolną liczbę rzeczywistą $x \in \mathbb{R}$ w dziedzinie liczb dualnych można przedstawić jako $(x, 0)$. Stąd, mnożenie liczby dualnej przez skalar $k \in \mathbb{R}$ to $k * (a, a') = (ka, ka')$. Można udowodnić, że liczby dualne tworzą dwuwymiarową algebrę nad ciałem liczb rzeczywistych. Poza możliwością zapisu wartości, do stworzenia praktycznej implementacji różniczkowania automatycznego potrzebne będą jeszcze funkcje, a dokładniej ich reprezentacja w dziedzinie liczb dualnych.

Przekształcenie $R \rightarrow D$

Rozszerzenie dziedziny popularnych funkcji na zbiór liczb dualnych jest bardzo proste. Rozważmy rozwinięcie rzeczywistej funkcji $f(a)$ w szereg Taylora

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(x_0)}{n!} (x - x_0)^n$$

Po podstawieniu $x = a + \epsilon a'$, $x_0 = a$ otrzymujemy

$$f(a + \epsilon a') = \sum_{n=0}^{\infty} \frac{f^n(a)}{n!} (a + \epsilon a' - a)^n = \sum_{n=0}^{\infty} \frac{f^n(a)(a')^n \epsilon^n}{n!} = f(a) + \epsilon a' f'(a)$$

Zauważ, że dla $n \geq 2$: $\epsilon^n = \epsilon^2 \epsilon^{n-2} = 0 * \epsilon^{n-2} = 0$

Przekształcenia dla popularnych funkcji

Tabela: Reprezentacja popularnych funkcji w dziedzinie liczb dualnych

Wartość dla $x \in \mathbb{R}$	Wartość w dziedzinie liczb dualnych
Wielomian $f(x) = \sum_{i=0}^n a_i x^i$	$f(x) = \sum_{i=0}^n a_i x^i + \epsilon x' \sum_{i=1}^n a_i i x^{i-1}$
e^x	$e^x + \epsilon x' e^x$
$\ln(x)$	$\ln(x) + \epsilon \frac{x'}{x}$
$\sin(x)$	$\sin(x) + \epsilon x' \cos(x)$
$\cos(x)$	$\cos(x) - \epsilon x' \sin(x)$
$\operatorname{tg}(x)$	$\operatorname{tg}(x) + \epsilon x' \sec^2(x)$
$\operatorname{ctg}(x)$	$\operatorname{ctg}(x) - \epsilon x' \csc^2(x)$
a^x	$a^x + \epsilon x' a^x \ln(a)$
$\sqrt{x}$	$\sqrt{x} + \epsilon \frac{x'}{2\sqrt{x}}$
c	$c + \epsilon 0 = c$

Prosta implementacja liczb dualnych w Haskellu

```
import Prelude as Pl

data D a = D (Double, Double) deriving (Eq, Show)

instance (Num a) => Num (D a) where
    D (a,b) + D (c,d) = D (a+c, b+d)
    D (a,b) * D (c,d) = D (a*c, a*d+b*c)
    D (a,b) - D (c,d) = D (a-c, b-d)
    fromInteger i = D (fromIntegral i, fromInteger 0)
    abs (D (a,b)) = D (abs a, abs b)
    signum (D (a,b)) = D (signum a, signum b)

instance (Fractional a) => Fractional (D a) where
    D (a,b) / D (c,d) = D (a/c, (b*c-a*d)/c*c)
    fromRational i = D (fromRational i, 0)

instance (Floating a) => Floating (D a) where
    pi = D(pi, 0)
    sin (D (a,b)) = D(Pl.sin a, b*(Pl.cos a))
    cos (D (a,b)) = D(Pl.cos a, -b*Pl.sin a)
    tan (D (a,b)) = D(Pl.tan a, b/((Pl.cos a)**2))
    exp (D (a,b)) = D(Pl.exp a, b*Pl.exp a)
    D (a,b) ** D(c, _) = D(a ** c, b*c*(a**(c-1)))
```

Inne metody implementacji

- Przeciążanie operatorów,
- Przekształcenie kodu źródłowego.

Dziękuję za uwagę