Автоматическое дифференцирование в машинном обучении

Подходы к численному дифференцированию сложных функций

В компьютерной алгебре существует три основных подхода к решению задачи дифференцирования сложных функций :

1. Символьное дифференцирование

Автоматический вывод алгебраических выражений для получения формулы производной. Может быть реализовано в таких пакетах ПО, как Mathematica, Maple, SymPy, SagaMath.

```
In[•]:= f[x_] := Exp[x^2];
        f'[x]
        N[f'[2]]
```

Out[•]= $2\, e^{w0^2}\, w0$

Out[•]= 218.393

```
In[•]:= g[x_] := Sin[Tanh[Cos[x^3] + Sin[x^2 + 1]]];
        g'[x]
```

Out[•]= $\text{Cos}\left[\text{Tanh}\left[\text{Cos}\left[x^3\right] + \text{Sin}\left[1 + x^2\right]\right]\right] \text{Sech}\left[\text{Cos}\left[x^3\right] + \text{Sin}\left[1 + x^2\right]\right]^2 \left(2\, x \,\text{Cos}\left[1 + x^2\right] - 3\, x^2 \,\text{Sin}\left[x^3\right]\right)$

Плюсы: Выражение для вычисления может быть получено явно в виде выражения, вычисление которого может быть оптимизировано.

Минусы: Для сложных функций с большим количеством составных элементов не представляется возможным программная реализация метода.

2. Численное дифференцирование

Приближенное вычисление производной методом конечных разностей.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

Плюсы: основано на классических методах, алгоритмически не сложно.

Минусы: существуют проблемы накопления ошибок в результате итеративного вычисления с округлением в компьютерной арифметике.

3. Автоматическое дифференцирование (АД)

Автоматическая генерация кода, дополняющего основной код вычислений выражений. Сгенерированный код вычисляет производную функции по правилам рекурсивного разложения производной сложной функции на простые составляющие. Базовый принцип АД - декомпозиция дифференциала с использованием правила дифференцирования сложной функции (цепного правила).

Пример:

In[]:=
```
l[x_] := x^2 + 1;
h[x_] := Sin[x] + x;
g[x_] := 2 x + 1;
Simplify[D[g[h[l[x]]], {x, 2}]]
```

Out[]= $4 \left(1 + \text{Cos}\left[1 + x^2\right] - 2 x^2 \text{Sin}\left[1 + x^2\right]\right)$

Может быть записано следующим образом :

In[]:=
```
ClearAll[l, h, g, x, lp, hp, gp];
l'[x_] := lp
h'[x_] = hp;
g'[x_] = gp;
D[g[h[l[x]]], x]
```

Out[]= gp hp lp

Построение вычислений в АД

Существует два основных режима вычисления проиизводной в АД: прямое и обратное накопление значений. Прямое накопление начинает вычисления от **lp** и заканчивает **gp**. В префиксной скобочной записи функций это означает вычисления в порядке от функций младшего порядка (в смысле композиции) к более высоким порядкам. В обратном режиме производные вычисляются от функций более высокого порядка к функциям младших порядков композиции.

Сейчас, в компьютерных науках для вычисления производных сложных функций используется специальная арифметика дуальных чисел. В этой арифметике множество элементов - пары

In[]:=
```
{a, a'};
```

где a, a' - действительные числа, а основные операции над ними выглядят следующим образом:

ϵ^2 = 0
$(a + a' \epsilon) (b + b' \epsilon) = a b + (a b' + b a') \epsilon$
$(a+a'\epsilon)/(b+b'\epsilon)=a/(b+b'\epsilon)+a'\epsilon/(b+b'\epsilon)$
$(a + a' \epsilon) + (b + b' \epsilon) =a+b+(a'+b')\epsilon$
$(a + a' \epsilon) - (b + b' \epsilon)=a-b+(a'-b')\epsilon$

Т.о. в этой арифмете может быть вычислен любой полином по следующей формуле:

$$P(x + x' \epsilon) = p_0 + p_1(x + x' \epsilon) + \ldots + p_n(x + x' \epsilon)$$
$$= p_0 + p_1 x + \ldots + p_n x^n + p_1 x' \epsilon + 2 p_2 x x' \epsilon + \ldots + n p_n x^{n-1} x' \epsilon$$
$$= P(x) + P^{(1)}(x) x' \epsilon$$

Где $P^{(1)}$ обозначает производную P по x, а x' названется “зерном” и может быть выбрано по обстоятельствам.
Вычисления производятся над парами {a,a'}, с применением стандартной арифметики по первому элементу и с применением дифференциальной арифметики по второму элементу,

как описано выше.

Формулы простых вычислений в этой арифметике:
https://en.wikipedia.org/wiki/Automatic_differentiation#Automatic _differentiation _using _dual _numbers