Automatic Differentiation

Suma Dodmani

Why Automatic Differentiation

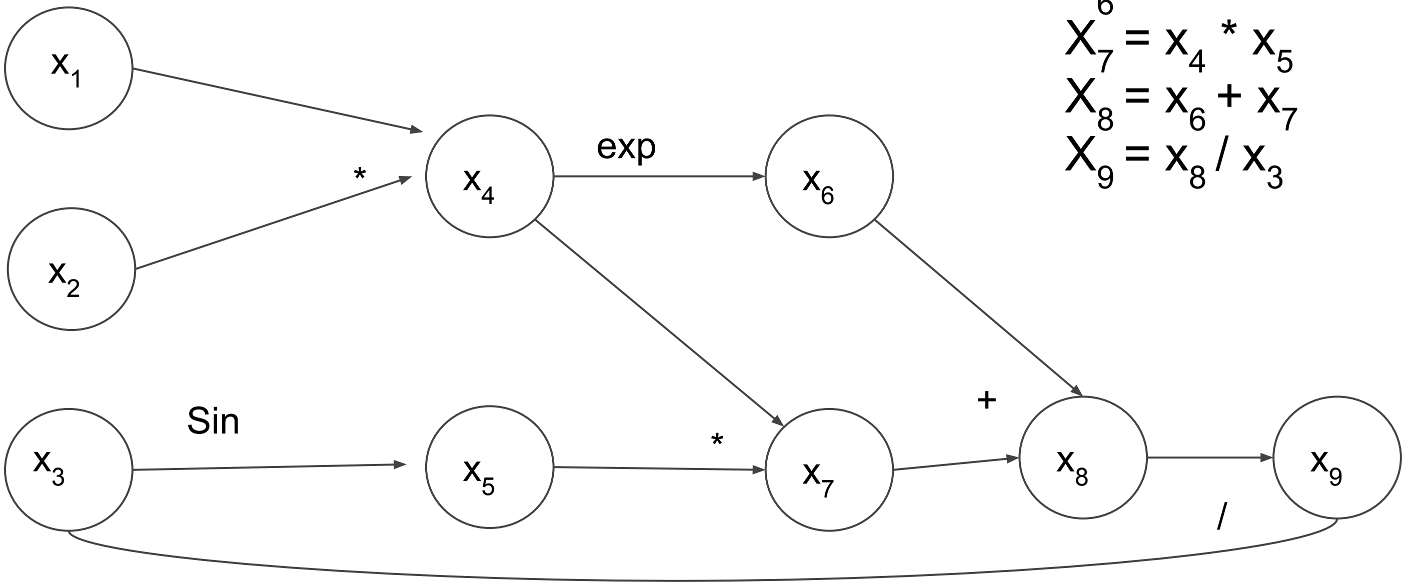
- Several numerical algorithms require the computation of derivatives.
    - Minimize objective function
    - To solve nonlinear system of equations
    - Systems of differential equations

Big Idea: Calculate a function’s rate of change and its value all at once.

Understanding Computational Graphs

$$f(x) = (x_1 x_2 sin x_3 + e^{x1x2})/x_3$$

$X_4 = x_1 * x_2$
$x_5 = \sin x_3$
$X_6 = e^{x4}$
$X_7 = x_4 * x_5$
$X_8 = x_6 + x_7$
$X_9 = x_8 / x_3$



Modes

- Forward Mode

  Evaluate and carry forward a directional derivative of each intermediate variable

- Reverse Mode

  Reverse sweep of the computational graph

The Forward Mode

Z = x * y + *sin(x)*

x = ?

y = ?

a = x * y

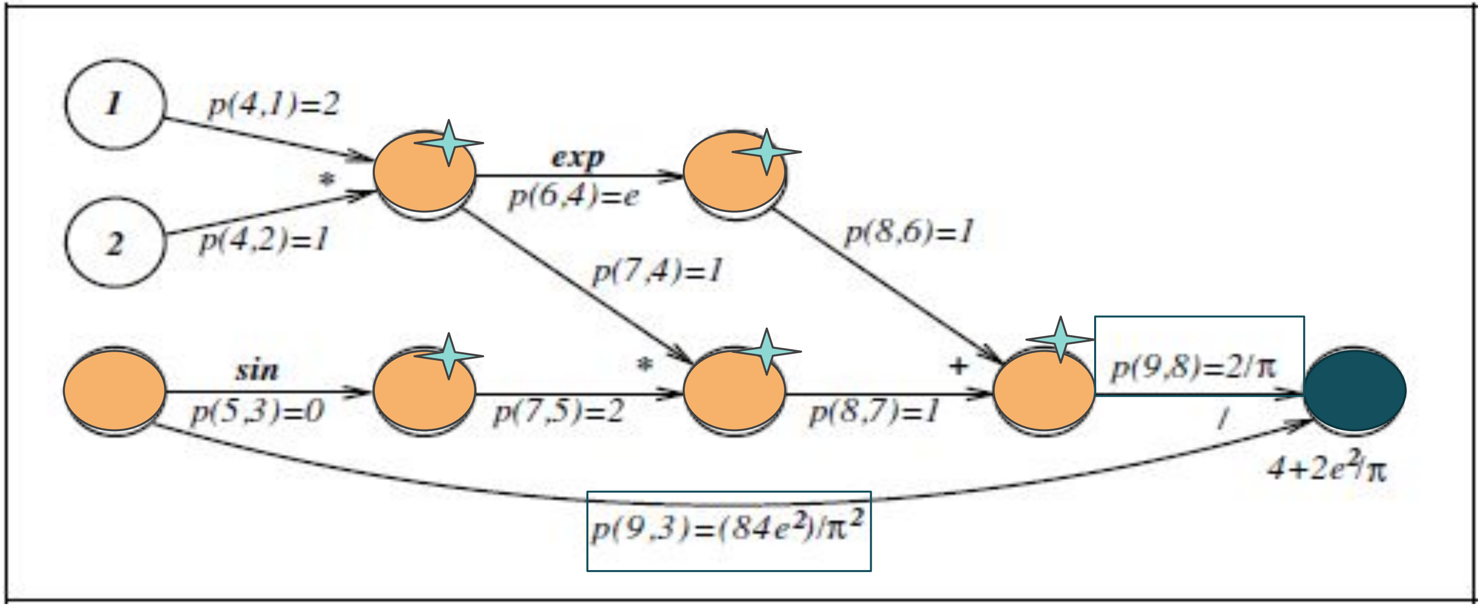
b = sin(x)

z = a + b

$$\frac{\partial w}{\partial t} = \sum_i \left(\frac{\partial w}{\partial u_i} \cdot \frac{\partial u_i}{\partial t} \right)$$
$$= \frac{\partial w}{\partial u_1} \cdot \frac{\partial u_1}{\partial t} + \frac{\partial w}{\partial u_2} \cdot \frac{\partial u_2}{\partial t} + \cdots$$

$$\frac{\partial x}{\partial t} = ?$$
$$\frac{\partial y}{\partial t} = ?$$
$$\frac{\partial a}{\partial t} = y \cdot \frac{\partial x}{\partial t} + x \cdot \frac{\partial y}{\partial t}$$
$$\frac{\partial b}{\partial t} = \cos(x) \cdot \frac{\partial x}{\partial t}$$
$$\frac{\partial z}{\partial t} = \frac{\partial a}{\partial t} + \frac{\partial b}{\partial t}$$

Reverse mode



Reverse Mode(Continued)

- The extra aritl ... ı is four or five times the arithmetic
- In forward mc ... ompute the gradient than to compu
- Drawback: St ... everse sweep.

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} = \nabla f(x) = \begin{bmatrix} (4+4e^2)/\pi \\ (2+2e^2)/\pi \\ (-8-4e^2)/\pi^2 \end{bmatrix},$$

Vector functions and Partial Separability

Partially Separable: A function *f is* called partially separable if there exists a family of n x n diagonal matrices U_i such that the function f can be represented as

$$f(x) = \sum f_i(U_i x)$$

- Many residual functions are partially separable as they are sum of residuals from various components of the model.

$$\nabla f(x) = J(x)^T e$$

Vector functions and Partial Separability

- **Constrained optimization:** Evaluating the objective function and constraint functions simultaneously.
- Exploring the shared intermediate nodes to reduce the workload.

Calculating Jacobians of Vector functions (Reverse Mode)

- Choose seed vector and applying the reverse mode to the scalar functions to get vector.

$$\nabla[r(x)^T q] = \nabla\left[\sum_{j=1}^{m} q_j r_j(x)\right] = J(x)^T q.$$

- **Forward Mode:** Jacobian Vector Product
- **Reverse Mode:** Jacobian-transpose-vector product

Calculating Hessians: Forward mode

$$D_{pq} x_i = p^T (\nabla^2 x_i) q,$$

- We define another scalar quantity for each node i in the computational graph.
- Evaluate these values during forward sweep alongside the function values and the first derivatives.
- Total increase factor for the number of arithmetic operations compared with the amount of arithmetic to evaluate f alone is a multiple of:

$$1 + n + N_z(\nabla^2 f),$$

Calculating Hessians: Reverse mode

Forward Mode : Evaluation of function and the gradient.

Reverse Mode: We apply the reverse sweep to the computed function. At the end of the reverse sweep, the nodes of the computation graph will have the values:

$$\frac{\partial}{\partial x_i}(\nabla f(x)^T q) = \left[\nabla^2 f(x) q\right]_i, \quad i = 1, 2, \ldots, n.$$

Conclusion

- Enhances Optimization algorithms
- Helps extract more information from the results of the computation

Thank You