自动微分模式

ZOMI 酱

Building a better connected world

www.mindspore.cn

关于本课程

1. 课程背景

- AI框架中自动微分的重要性

2. 课程内容

- 微分基本概念：数值微分 - 符号微分 - 自动微分
- 自动微分模式：前向微分 – 后向微分 – 雅克比原理
- 具体实现方式：表达式或图 – 操作符重载OO – 源码转换 AST
- MindSpore实现：基于图表示的源码转换Graph Base AST
- 自动微分的未来
- 自动微分的挑战

What is AD

自动微分：所有数值计算都由有限的基本运算组成

基本运算的导数表达式是已知的

通过链式法则将数值计算各部分组合成整体

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

Forward Primal Trace

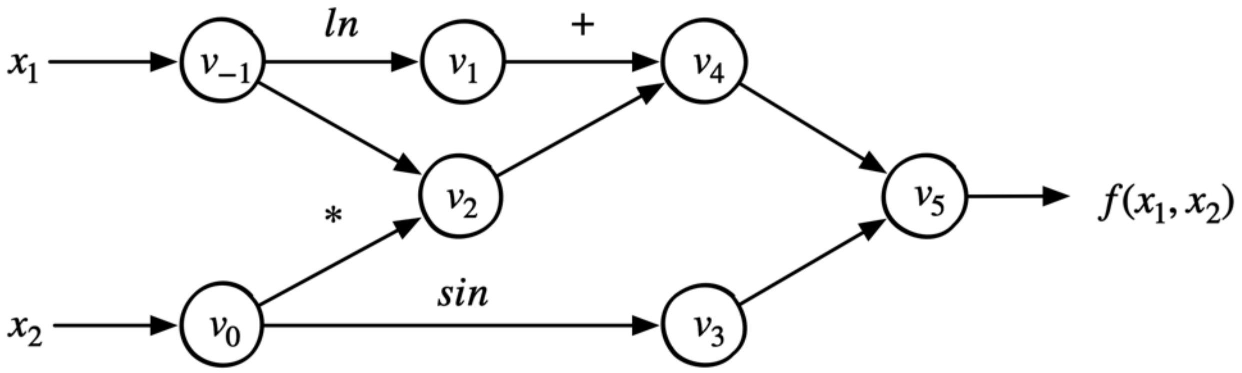
v_{-1}	$= x_1$	$= 2$
v_0	$= x_2$	$= 5$
v_1	$= \ln v_{-1}$	$= \ln 2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
v_3	$= \sin v_0$	$= \sin 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$
y	$= v_5$	$= 11.652$

What is AD

原函数：

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

原函数转换成DAG（有向无环图）：



根据链式求导法则展开：

$$\frac{\partial f}{\partial x_1} = \frac{\partial v_{-1}}{\partial x_1}\left(\frac{\partial v_1}{\partial v_{-1}}\frac{\partial v_4}{\partial v_1} + \frac{\partial v_2}{\partial v_{-1}}\frac{\partial v_4}{\partial v_2}\right)\frac{\partial v_5}{\partial x_4}\frac{\partial f}{\partial v_5}$$

AD Forward Mode

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

$$\dot{v}_i = \frac{\partial v_i}{\partial x_1} \qquad \dot{y}_j = \frac{\partial y_j}{\partial x_i}$$

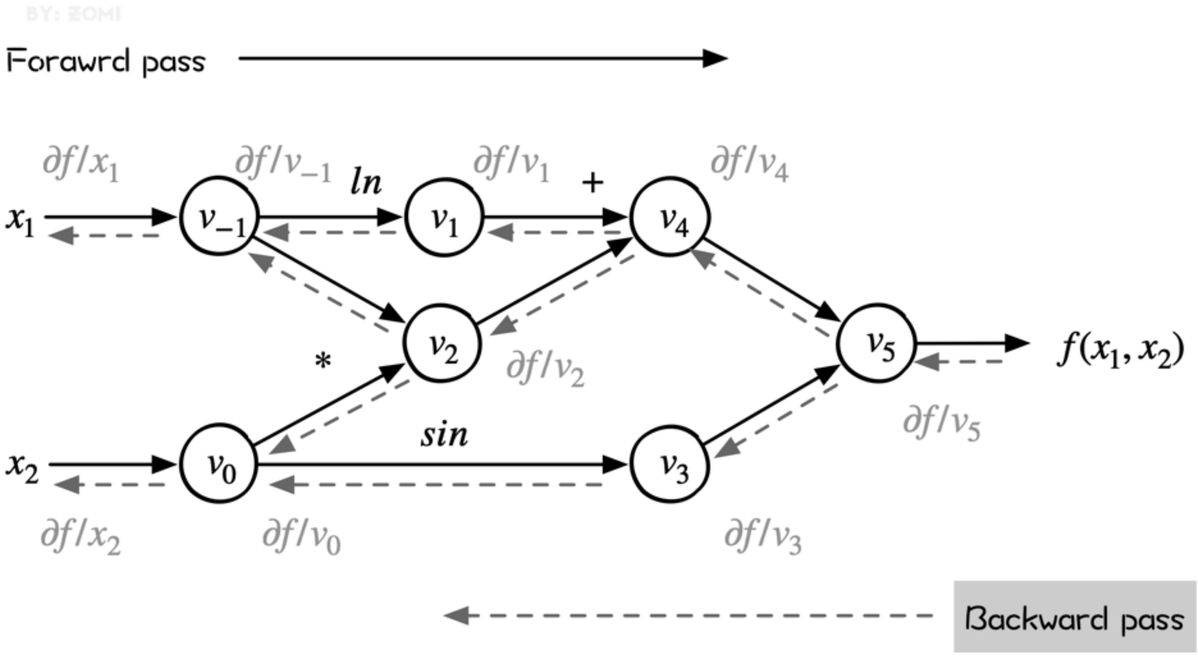
Forward Primal Trace			Forward Tangent (Derivative) Trace		
v_{-1}	$= x_1$	$= 2$	$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
v_0	$= x_2$	$= 5$	$\dot{v}_0$	$= \dot{x}_2$	$= 0$
v_1	$= \ln v_{-1}$	$= \ln 2$	$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$	$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1 \times 5 + 0 \times 2$
v_3	$= \sin v_0$	$= \sin 5$	$\dot{v}_3$	$= \dot{v}_0 \times \cos v_0$	$= 0 \times \cos 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$	$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$	$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
y	$= v_5$	$= 11.652$	$\boldsymbol{\dot{y}}$	$\boldsymbol{= \dot{v}_5}$	$\mathbf{= 5.5}$

What is AD

原函数：

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

原函数转换成DAG：



根据链式求导法则展开：

$$\frac{\partial f}{\partial \boldsymbol{x}} = \sum_{k=1}^{N} \frac{\partial f}{\partial v_k} \frac{\partial v_k}{\partial \boldsymbol{x}}$$

AD Reverse Mode

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

$$\bar{v}_i = \frac{\partial y_i}{\partial v_i}$$

Forward Primal Trace

$v_{-1} = x_1$		$= 2$
$v_0 = x_2$		$= 5$
$v_1 = \ln v_{-1}$		$= \ln 2$
$v_2 = v_{-1} \times v_0$		$= 2 \times 5$
$v_3 = \sin v_0$		$= \sin 5$
$v_4 = v_1 + v_2$		$= 0.693 + 10$
$v_5 = v_4 - v_3$		$= 10.693 + 0.959$
$y = v_5$		$= 11.652$

Reverse Adjoint (Derivative) Trace

$\bar{x}_1 = \bar{v}_{-1}$		$= 5.5$
$\bar{x}_2 = \bar{v}_0$		$= 1.716$
$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$	$= 5.5$
$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$	$= 1.716$
$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$	$= 5$
$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$	$= -0.284$
$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$	$= -1$
$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$	$= 1$
$\bar{v}_5 = \bar{y}$	$= 1$	

Jacobian Matrix

对于函数 $\vec{\boldsymbol{y}} = \boldsymbol{f}(\vec{\boldsymbol{x}})$ ，其中 $\boldsymbol{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$，那么 $\vec{\boldsymbol{y}}$ 中关于 $\vec{\boldsymbol{x}}$ 的梯度可以表示为 Jacobian 矩阵：

$$\boldsymbol{J}_f = \begin{bmatrix} \frac{\partial \boldsymbol{y}}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{y}}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

AD Forward Mode: Jacobian-Vector Production

对于函数 $\vec{\boldsymbol{y}} = \boldsymbol{f}(\vec{\boldsymbol{x}})$ ，其中 $\boldsymbol{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，那么 $\vec{\boldsymbol{y}}$ 中关于 $\vec{\boldsymbol{x}}$ 的梯度可以表示为 Jacobian 矩阵：

$$\boldsymbol{J}_f = \begin{bmatrix} \frac{\partial \boldsymbol{y}}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{y}}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

设置 $\vec{\boldsymbol{v}}$ 是关于函数 $l = g(\vec{\boldsymbol{y}})$ 的梯度：

$$\vec{\boldsymbol{v}} = \begin{bmatrix} \frac{\partial l}{\partial y_1} & \cdots & \frac{\partial l}{\partial y_m} \end{bmatrix}^{\mathrm{T}}$$

Jacobian - vector 积就是函数 l 中关于 x_1 的梯度：

$$\boldsymbol{J} \cdot \vec{\boldsymbol{v}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial l}{\partial y_1} \\ \vdots \\ \frac{\partial l}{\partial y_m} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \vdots \\ \frac{\partial y_m}{\partial x_1} \end{bmatrix}$$

AD Reverse Mode: Vector-Jacobian Production

对于函数 $\vec{\boldsymbol{y}} = \boldsymbol{f}(\vec{\boldsymbol{x}})$ ，其中 $\boldsymbol{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$，那么 $\vec{\boldsymbol{y}}$ 中关于 $\vec{\boldsymbol{x}}$ 的梯度可以表示为 Jacobian 矩阵：

$$\boldsymbol{J}_f = \begin{bmatrix} \frac{\partial \boldsymbol{y}}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{y}}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

设置 $\vec{\boldsymbol{v}}$ 是关于函数 $l = g(\vec{\boldsymbol{y}})$ 的梯度：

$$\vec{\boldsymbol{v}} = \begin{bmatrix} \frac{\partial l}{\partial y_1} & \cdots & \frac{\partial l}{\partial y_m} \end{bmatrix}^{\mathrm{T}}$$

vector – Jacobian 积就是函数 l 中关于 $\vec{\boldsymbol{x}}$ 的梯度：

$$\boldsymbol{J}^{\mathrm{T}} \cdot \vec{\boldsymbol{v}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial l}{\partial y_1} \\ \vdots \\ \frac{\partial l}{\partial y_m} \end{bmatrix} = \begin{bmatrix} \frac{\partial l}{\partial x_1} \\ \vdots \\ \frac{\partial l}{\partial x_n} \end{bmatrix}$$

Reverse Mode VS Forward Mode

对于函数 $f: \mathbb{R}^n \to \mathbb{R}^m$，有 Jacobian 矩阵：

$$J_f = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

对于一个输出变量 y_i 进行一次反向模式，迭代算出 Jacobian 矩阵每一行

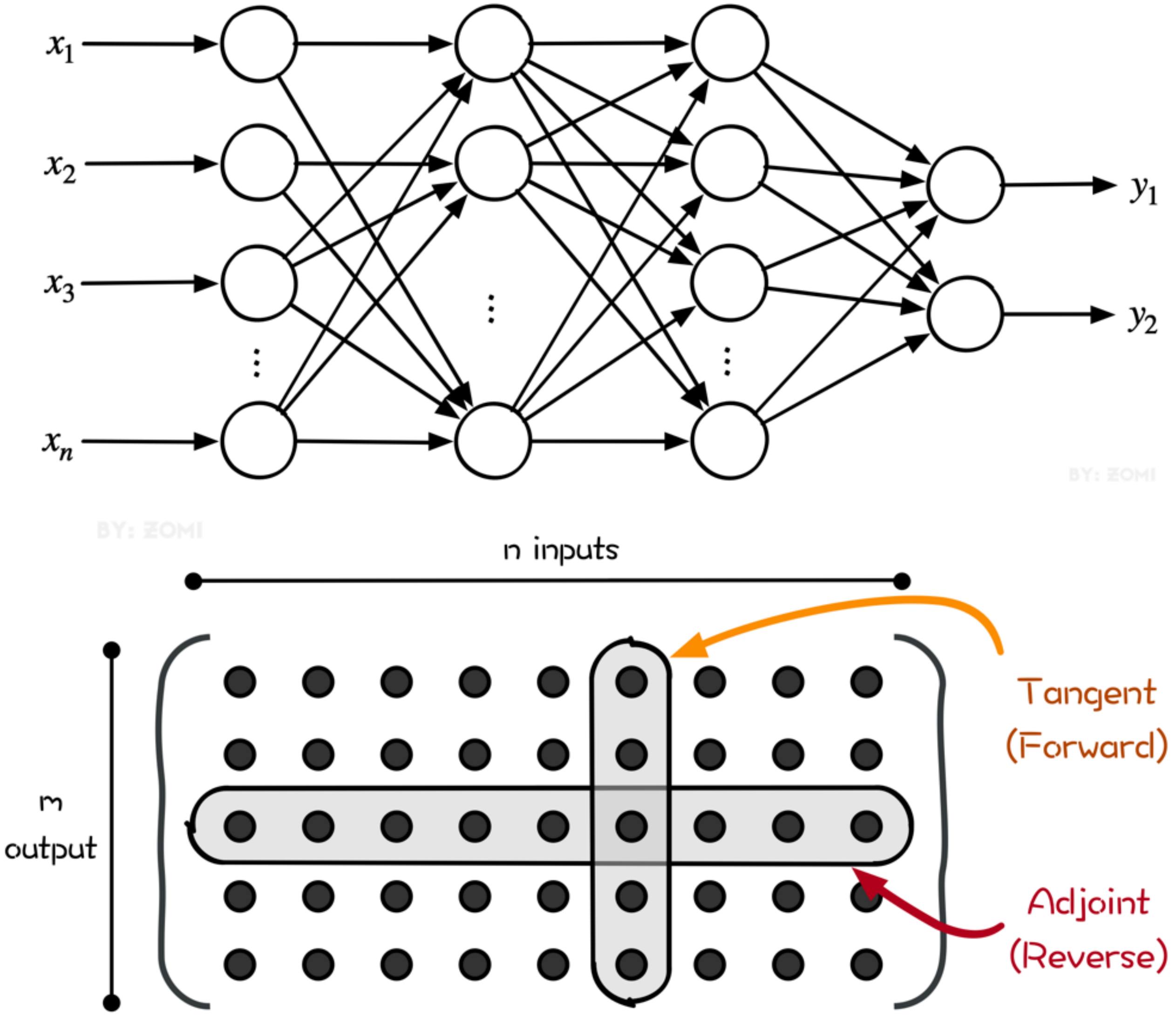
对于一个输入变量 x_i 进行一次前向模式，迭代算出Jacobian矩阵每一列

- 当 m > n，适合使用前向模式自动微分；
- 当 n > m，适合使用反向模式自动微分；

Automatic Differentiation in ML

Jacobian 矩阵：

$$J_f = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$



Automatic Differentiation

自动微分：所有数值计算都由有限的基本运算组成

基本运算的导数表达式是已知的

通过链式法则将数值计算各部分组合成整体

链式法则将结果，组合得到整体程序的求导结果：

$$(f \cdot g)'(x) = f'\big(g(x)\big)g'(x)$$

分为前向模式和反向模式，均为求解 Jacobian 矩阵：

$$\boldsymbol{J}_f = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

优势

- 数值精度高
- 无表达式膨胀

缺点

- 需要存储中间求导结果
- 占用大量计算机内存

Conclusion

1. 了解到自动微分分为前向微分和反向微分
2. 了解雅克比矩阵的基本原理和表示，前向和反向微分模式的雅克比表示
3. 了解了自动微分的优缺点和在AI框架中常用的基本模式

BUILDING A BETTER CONNECTED WORLD

THANK YOU