挑战与未来

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
   - 自动微分的挑战与未来

AD Challenge - Ease of use

- 理想中的自动微分是对**数学表达**的分解、微分和组合过程

$$l_1 = x$$
$$l_{n+1} = 4l_n(1 - l_1)$$
$$f(x) = l_4 = 64x(1-x)(1-2x)^2(1-8x+8x^2)^2$$

- 实际中的自动微分是对**程序表达**的分解、微分和组合过程

```
f(x):
    v = x
    for i = 1 to 3:
        v = 4 * v * (1 - v)
    return v
```

AD Challenge - Ease of use (I)

控制流表达问题

$l_1 = x$

$l_{n+1} = 4l_n(1 - l_1)$

$f(x) = l_4 = 64x(1-x)(1-2x)^2(1-8x+8x^2)^2$

```
f(x):
    v = x
    for i = 1 to 3:
        v = 4 * v * (1 - v)
    return v
```

识别程序表达中用于计算控制流的运算部分，并将其排除在微分过程外

复杂数据类型

$\mathrm{d}x = x$

$l_{n+1} = 4l_n(1 - l_1)$

$f(x) = l_4 = 64x(1-x)(1-2x)^2(1-8x+8x^2)^2$

```
<aexp> ::= NUMBER | STRING | VAR | BOOLEAN | PRIMOP

Python ::= [List, Enum, Tuple, Dict, DefaultDict]
C++ ::= [size_t, whcar_t, enum, struct , STL::list]
```

AD Challenge - Ease of use (II)

语言特性

- 多态、异常处理、调试、IO处理、继承等

需求重写

- 物理模拟、游戏引擎、气候模拟有DSL属性

AD Challenge - Performance

程序与微分表达

$$f = x^3$$

$$dx = 3 * x^2$$

```
def fun(x):
    t = x * x * x
    return t

def dfun(x):
    dx = 3 * x * x
    return dx
```

```
def fun(x):
    t = x * x
    v = x * t
    dx = 3 * t
    return v, dx
```

AD Challenge - Performance

额外中间变量

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

Forward Primal Trace			Reverse Adjoint (Derivative) Trace			
$v_{-1} = x_1$		$= 2$	$\bar{x}_1 = \bar{v}_{-1}$			$= 5.5$
$v_0 = x_2$		$= 5$	$\bar{x}_2 = \bar{v}_0$			$= 1.716$
$v_1 = \ln v_{-1}$		$= \ln 2$	$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$		$= 5.5$
$v_2 = v_{-1} \times v_0$		$= 2 \times 5$	$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$		$= 1.716$
			$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$		$= 5$
$v_3 = \sin v_0$		$= \sin 5$	$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$		$= -0.284$
$v_4 = v_1 + v_2$		$= 0.693 + 10$	$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$		$= 1$
			$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$		$= 1$
$v_5 = v_4 - v_3$		$= 10.693 + 0.959$	$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$		$= -1$
			$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$		$= 1$
$y = v_5$		$= 11.652$	$\bar{v}_5 = \bar{y}$	$= 1$		

AD Challenge - Performance

额外中间变量

二阶微分方程的一般形式：

$$F(x, y, y', y'') = 0$$

其中，x是自变量，y是未知函数，y'是y的一阶导数，y''是y的二阶导数。

$$\frac{d^2y}{dx^2} = \frac{d}{dy}\left(\frac{dy}{dx}\left(\frac{dx}{x}\right)\right)$$

AD Challenge - Performance

重计算与编译优化

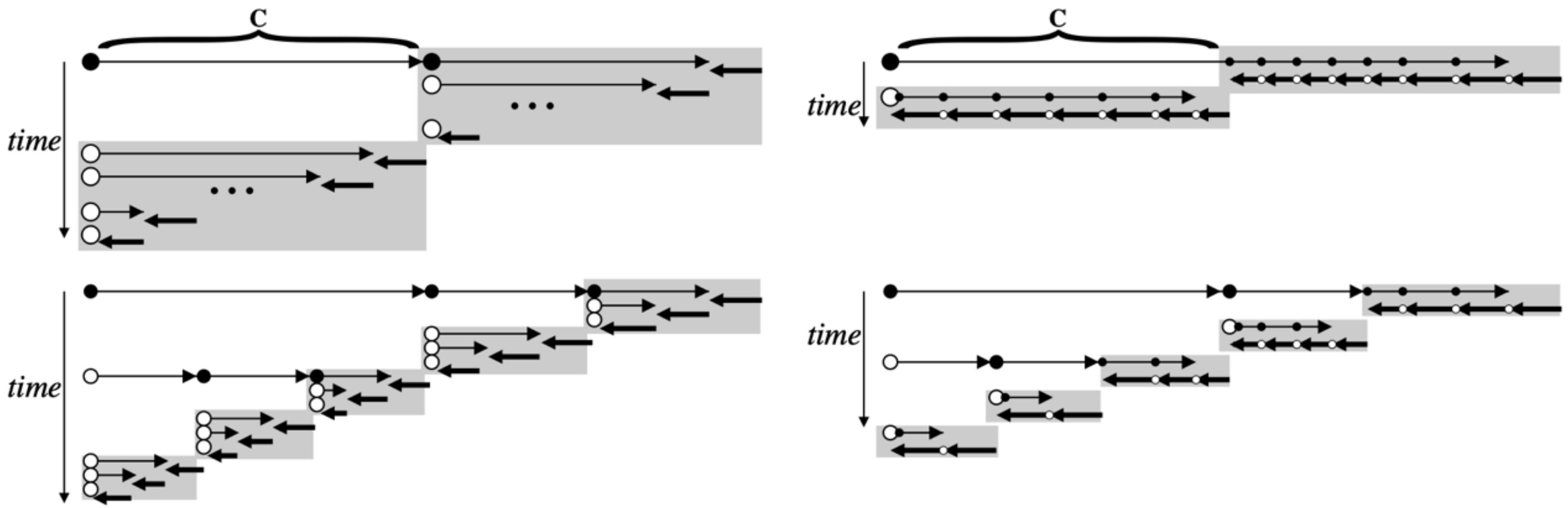

Figure 3: Checkpointing with the Recompute-All (*left*) and Store-All (*right*) approaches. The shaded areas reproduce the basic pattern of the chosen approach. *top:* single checkpointing, *bottom* nested checkpointing.

AD Future

可微编程

将自动微分技术与语言设计、编译器 / 解释器甚至 IDE 等工具链等深度融合，将微分作为语言中 first-class feature

Conclusion

1. 自动微分挑战主要集中在易用性和性能两方面
2. 易用性受限于控制流、数据类型等语言特性以外，还受限于领域需求
3. 性能主要以程序表达与微分表达结合，编译，甚至高阶微分等引起

BUILDING A BETTER CONNECTED WORLD

THANK YOU