Advanced Algorithmic Differentiation: Higher-order Derivatives

Andrea Walther and Sri Tadinada
Institut für Mathematik
Humboldt-Universität zu Berlin

CWI Autumn School 2025
Artificial Intelligence in Natural Sciences

Outline

Second-order Derivatives

Use forward and reverse mode AD together!

$$y = F(x) \quad \longrightarrow \quad \bar{x} = \bar{y}^\top F'(x)$$

Second-order Derivatives

Use forward and reverse mode AD together!

$$y = F(x) \longrightarrow \bar{x} = \bar{y}^\top F'(x)$$

$$\bar{x} = \bar{y}F'(x) \longrightarrow \dot{\bar{x}} = \bar{y}^\top F''(x)\dot{x} + \dot{\bar{y}}F'(x)$$

Second-order Derivatives

Use forward and reverse mode AD together!

$$y = F(x) \quad \longrightarrow \quad \bar{x} = \bar{y}^\top F'(x)$$

$$\bar{x} = \bar{y}F'(x) \quad \longrightarrow \quad \dot{\bar{x}} = \bar{y}^\top F''(x)\dot{x} + \dot{\bar{y}}F'(x)$$

For scalar-valued function $F, \bar{y} = 1, \dot{\bar{y}} = 0$, and $\dot{x}$ one has

$$\dot{\bar{x}} = F''(x)\dot{x} = \text{ Hessian-vector product}$$

Complexity (Second-order Adjoints)

grad	c	$\pm$	$*$	ψ
MOVES	$2+2$	$12+6$	$11+11$	$7+7$
ADDS	0	$3+3$	$2+5$	$1+2$
MULTS	0	0	$3+6$	$1+4$
NLOPS	0	0	0	$2+2$

$$\Rightarrow \boxed{\begin{array}{lcl} \mathrm{OPS}(\bar{y}^\top F''(x)\dot{x}) & \leq & c\,\mathrm{OPS}(F(x)) \\ \mathrm{MEM}(\bar{y}^\top F''(x)\dot{x}) & \sim & \mathrm{OPS}(F(x)) \end{array}}$$

with $c \in [7, 10]$ platform dependent

Higher-order Derivatives I

Think in Taylor polynomials!

Let x be the path

$$x(t) \quad \equiv \quad \sum_{j=0}^{d} x_j t^j \; : \; \mathbb{R} \; \mapsto \; \mathbb{R}^n \qquad \text{with} \qquad x_j \; = \; \frac{1}{j!} \, \frac{\partial^j}{\partial t^j} x(t) \Big|_{t=0}$$

Higher-order Derivatives I

Think in Taylor polynomials!

Let x be the path

$$x(t) \equiv \sum_{j=0}^{d} x_j t^j : \mathbb{R} \mapsto \mathbb{R}^n \qquad \text{with} \qquad x_j = \frac{1}{j!} \frac{\partial^j}{\partial t^j} x(t) \Big|_{t=0}$$

Hence

$$x_j \text{ is scaled derivative at } t = 0 \quad \Rightarrow \quad x_1 = \text{tangent}, \; x_2 = \text{curvature}$$

Higher-order Derivatives I

Think in Taylor polynomials!

Let x be the path

$$x(t) \equiv \sum_{j=0}^{d} x_j t^j : \mathbb{R} \mapsto \mathbb{R}^n \qquad \text{with} \qquad x_j = \frac{1}{j!} \frac{\partial^j}{\partial t^j} x(t) \Big|_{t=0}$$

Hence

$$x_j \text{ is scaled derivative at } t = 0 \quad \Rightarrow \quad x_1 = \text{tangent}, \; x_2 = \text{curvature}$$

Now:
Consider $F : \mathbb{R}^n \to \mathbb{R}^m$, d times continuously differentiable
$\Rightarrow y(t) \equiv F(x(t)) : \mathbb{R} \mapsto \mathbb{R}^m$ is smooth (chain rule!)

Higher-order Derivatives II

There exist Taylor coefficients $y_j \in \mathbb{R}^m$, $0 \leq j \leq d$, with

$$y(t) \;=\; \sum_{j=0}^{d} y_j t^j + O(t^{d+1}).$$

Higher-order Derivatives II

There exist Taylor coefficients $y_j \in \mathbb{R}^m$, $0 \leq j \leq d$, with

$$y(t) \quad = \quad \sum_{j=0}^{d} y_j t^j + O(t^{d+1}).$$

Using the chain rule, one obtains:

$$\begin{aligned}
y_0 &= F(x_0) \\
y_1 &= F'(x_0)\, x_1 \\
y_2 &= F'(x_0)\, x_2 + \tfrac{1}{2} F''(x_0)\, x_1 x_1 \\
y_3 &= F'(x_0)\, x_3 + F''(x_0)\, x_1 x_2 + \tfrac{1}{6} F'''(x_0)\, x_1 x_1 x_1 \\
&\dots
\end{aligned}$$

Higher-order Derivatives II

There exist Taylor coefficients $y_j \in \mathbb{R}^m$, $0 \leq j \leq d$, with

$$y(t) \quad = \quad \sum_{j=0}^{d} y_j t^j + O(t^{d+1}).$$

Using the chain rule, one obtains:

$$\begin{aligned}
y_0 &= F(x_0) \\
y_1 &= F'(x_0)\, x_1 \\
y_2 &= F'(x_0)\, x_2 + \tfrac{1}{2} F''(x_0)\, x_1 x_1 \\
y_3 &= F'(x_0)\, x_3 + F''(x_0)\, x_1 x_2 + \tfrac{1}{6} F'''(x_0)\, x_1 x_1 x_1 \\
&\dots
\end{aligned}$$

$\Rightarrow$ directional derivatives of high order (forward mode)

Univariate Taylor Mode of AD

The propagation of

$$x_0, \ldots, x_d \to y_0, \ldots, y_d$$

is called univariate Taylor mode of AD

Univariate Taylor Mode of AD

The propagation of

$$x_0, \ldots, x_d \to y_0, \ldots, y_d$$

is called univariate Taylor mode of AD

Implementation?

- naive way: qubic computational cost in highest degree d
- Faà Di Bruno (1857!) proposed adapted approach with quadratic computational cost in highest degree d

Univariate Taylor Mode of AD

The propagation of

$$x_0, \ldots, x_d \rightarrow y_0, \ldots, y_d$$

is called univariate Taylor mode of AD

Implementation?

- naive way: qubic computational cost in highest degree d
- Faà Di Bruno (1857!) proposed adapted approach with quadratic computational cost in highest degree d

Hence:

$$\mathrm{OPS}(x_0, \ldots, x_d \rightarrow y_0, \ldots, y_d) \sim d^{\,2}\,\mathrm{OPS}(x_0 \rightarrow y_0)$$

using suitable Taylor arithmetic

Univariate Taylor Mode: Example

From

$$\begin{aligned}
y_0 &= F(x_0) \\
y_1 &= F'(x_0)\,x_1 \\
y_2 &= F'(x_0)\,x_2 + \tfrac{1}{2}F''(x_0)\,x_1x_1 \\
y_3 &= F'(x_0)\,x_3 + F''(x_0)\,x_1x_2 + \tfrac{1}{6}F'''(x_0)\,x_1x_1x_1
\end{aligned}$$

it follows for $x_1 = 1, x_2 = 0, x_3 = 0$ that

$$\begin{aligned}
y_0 &= F(x_0) \\
y_1 &= F'(x_0)\,x_1 \\
y_2 &= \tfrac{1}{2}F''(x_0)\,x_1x_1 \\
y_3 &= \tfrac{1}{6}F'''(x_0)\,x_1x_1x_1
\end{aligned}$$

i.e., scaled Taylor coefficients of F!

Physics-Informed Neural Networks (PINNs)

Idea:
PINNs integrate *physical laws* (e.g. PDEs) into the training process of neural networks.

Goal:
Approximate a solution $u_\theta(x, t)$ of a PDE $\mathcal{N}[u_\theta(x, t)] = 0$, by minimizing

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}}.$$

Physics-Informed Neural Networks (PINNs)

Idea:
PINNs integrate *physical laws* (e.g. PDEs) into the training process of neural networks.

Goal:
Approximate a solution $u_\theta(x, t)$ of a PDE $\mathcal{N}[u_\theta(x, t)] = 0$, by minimizing

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}}.$$

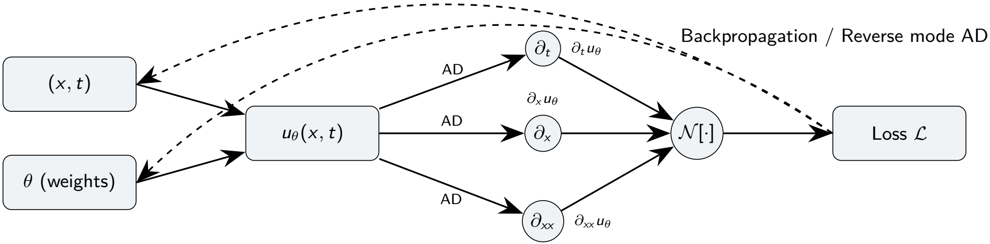
Input (x, t) → Neural Network $u_\theta(x, t)$ → PDE Residual $\mathcal{N}[u_\theta]$ → Loss $\mathcal{L}$

Key advantage: Leverages known physics to reduce data demand and improve generalization.

Derivatives for PINNs I

PINNs need spatial/temporal derivatives to inserted them into the PDE operator $\mathcal{N}$

$$\mathcal{N}[u_\theta] = \frac{\partial u_\theta}{\partial t} - \nu \frac{\partial^2 u_\theta}{\partial x^2} + u_\theta \frac{\partial u_\theta}{\partial x} = 0$$



Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)

   then

2. backpropagation (=reverse mode AD) applied to these derivatives

Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)
   approaches:
   - nested AD
   - univariate Taylor approach (sketched before)
   - multivariate Taylor approach

   then

2. backpropagation (=reverse mode AD) applied to these derivatives

Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)
   approaches:
   - nested AD
     exponential complexity
   - univariate Taylor approach (sketched before)
   - multivariate Taylor approach

   then

2. backpropagation (=reverse mode AD) applied to these derivatives

Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)
   approaches:
   - nested AD
     exponential complexity
   - univariate Taylor approach (sketched before)
     complexity grows quadratically in the degree of the PDE
   - multivariate Taylor approach

   then

2. backpropagation (=reverse mode AD) applied to these derivatives

Berlin Mathematics Research Center

Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)
   approaches:
   - nested AD
     exponential complexity
   - univariate Taylor approach (sketched before)
     complexity grows quadratically in the degree of the PDE
   - multivariate Taylor approach
     object of current research

   then

2. backpropagation (=reverse mode AD) applied to these derivatives

Derivatives for PINNs II

Hence

1. calculation of derivatives for PDE (higher order!)
   approaches:
   - nested AD
     exponential complexity
   - univariate Taylor approach (sketched before)
     complexity grows quadratically in the degree of the PDE
   - multivariate Taylor approach
     object of current research

   then

2. backpropagation (=reverse mode AD) applied to these derivatives
   $\Rightarrow$ order $=$ order of PDE $+ 1$!

Collapsed Taylor Mode

F. Dangel, T. Siebert, M. Zeinhofer, A. Walther:
Collapsing Taylor Mode Automatic Differentiation (to appear in NEURIPS proceedings 2025)

Observation: PDE operators often contain sum of derivatives, e.g., Laplacian

Collapsed Taylor Mode

F. Dangel, T. Siebert, M. Zeinhofer, A. Walther:
Collapsing Taylor Mode Automatic Differentiation (to appear in NEURIPS proceedings 2025)

Observation: PDE operators often contain sum of derivatives, e.g., Laplacian

Idea: Exploit linearity in highest derivative,
i.e., propagate sum, instead of summing up at the end!
Then: Trace is cheaper than complete Hessian

Collapsed Taylor Mode

F. Dangel, T. Siebert, M. Zeinhofer, A. Walther:
Collapsing Taylor Mode Automatic Differentiation (to appear in NEURIPS proceedings 2025)

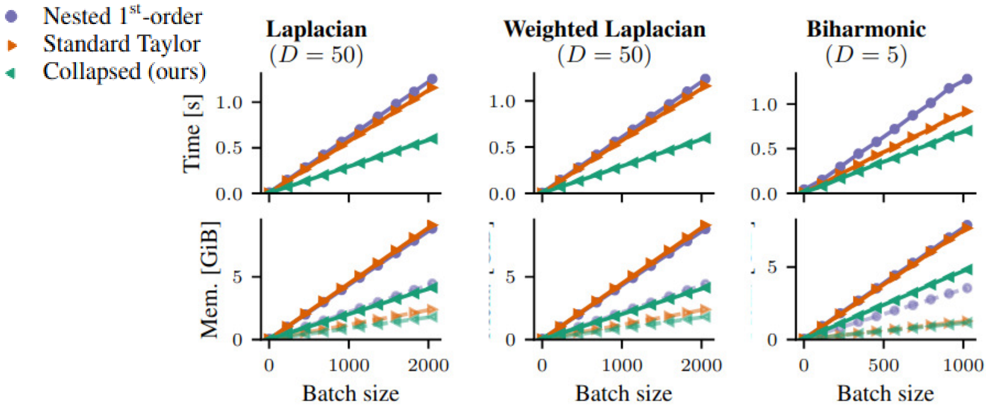
Observation: PDE operators often contain sum of derivatives, e.g., Laplacian

Idea: Exploit linearity in highest derivative,
i.e., propagate sum, instead of summing up at the end!
Then: Trace is cheaper than complete Hessian

Computational complexity for space dimension n:

$$\begin{array}{ll} \text{univariate:} & 1+n+n \text{ vectors} \\ \text{collapsed:} & 1+n+1 \text{ vectors} \end{array}$$

Collapsed Taylor Mode (Example)



Summary

- Higher-order derivatives important for PINNS and Variational Monte Carlo Simulations
- Various approaches:
  Nesting
  (Collapsed) Taylor mode
  multivariate method
- PyTorch implementation of collpased Taylor mode available at https://github.com/f-dangel/torch-jet