NASA-STIG — AD / PyTorch / JAX

Accelerated Computation with Auto-differentiation

Phill Cargile
Research Scientist
Center for Astrophysics | Harvard & Smithsonian
Dec. 15th 2025

Outline:

- Autodiff Motivation: Why do we need anything beyond NumPy & SciPy?
- PyTorch (Dec. 15th)
  - Model Building
  - Training
- JAX + JAX Ecosystem (Dec. 22nd)

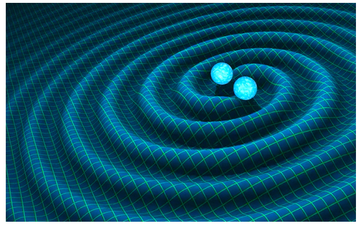
NumPy & SciPy

Tasks:

- Efficient N-D Array Math
- Vectorized Operations
- Broadcasting
- Statistical Functions
- Linear Algebra
- FFT
- Random Numbers

Tasks:

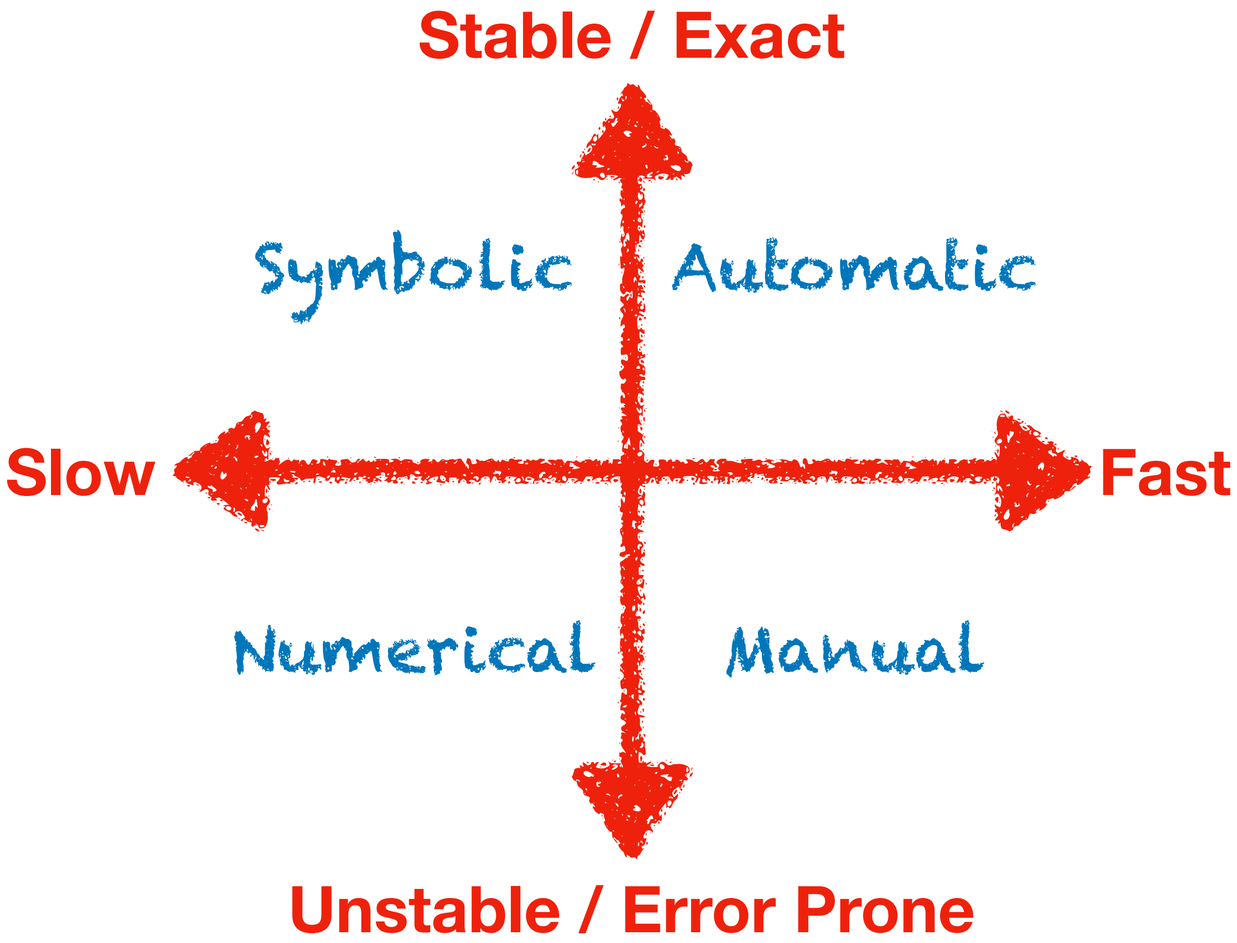
- Regression / Optimization
- Signal Processing
- Interpolation
- Integration



The Astropy Project

What is the partial derivative of a Python function?

Ways to compute a derivative:

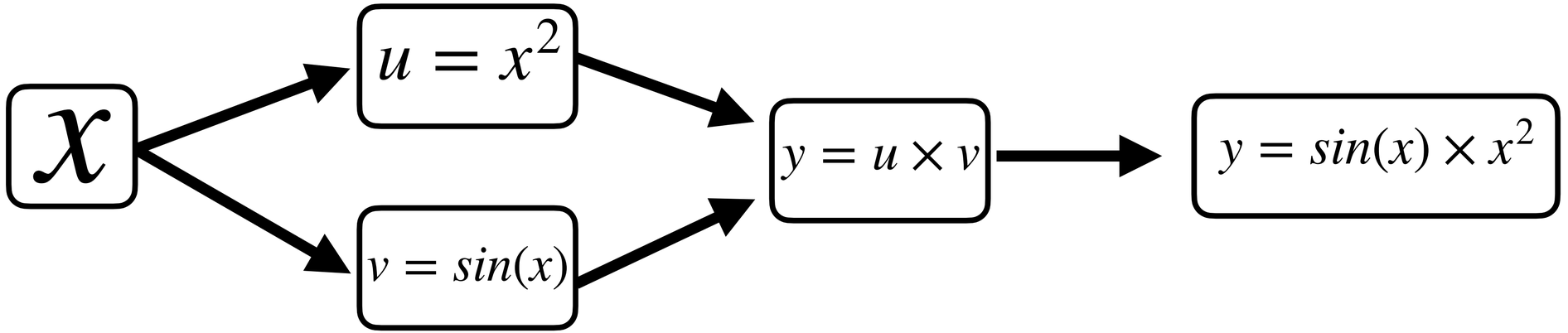


Automatic Differentiation (Autodiff, AD, etc.)

i.e., the chain-rule!

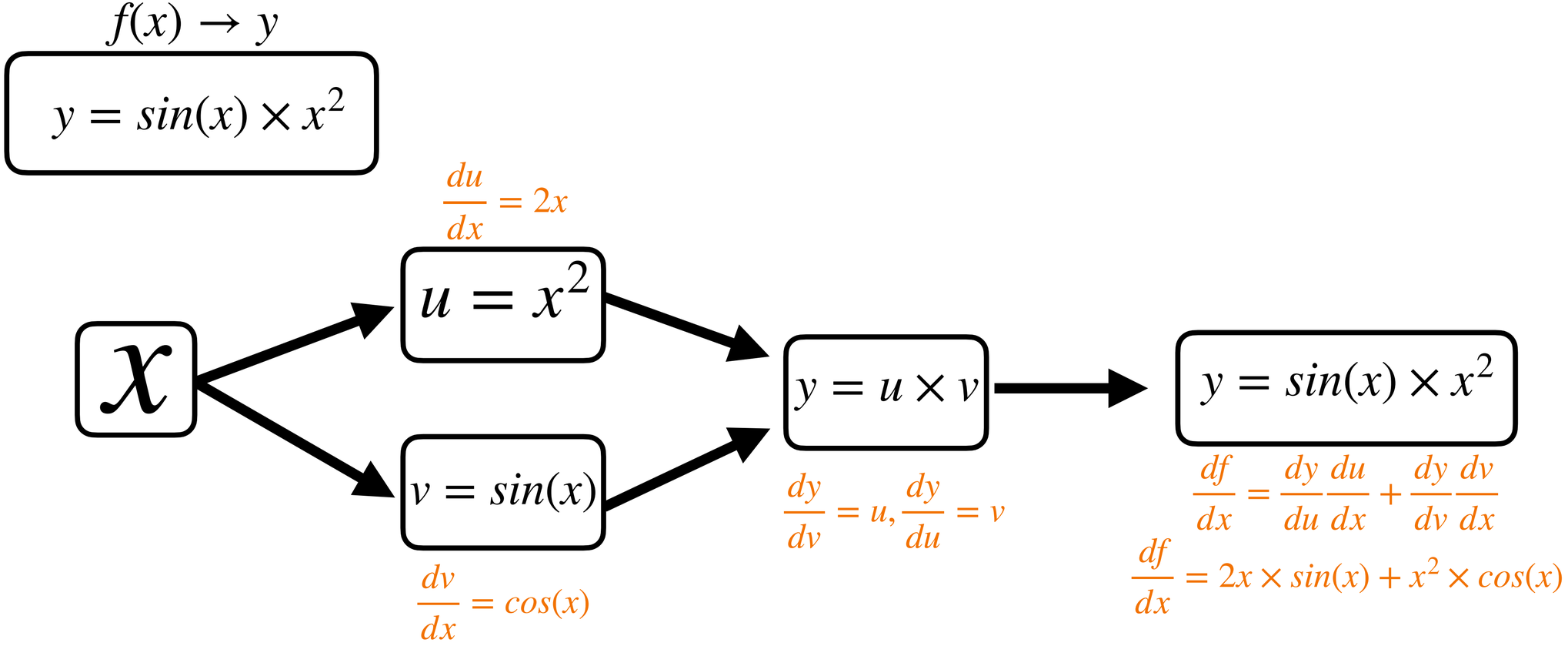
$f(x) \rightarrow y$

$$y = sin(x) \times x^2$$



Automatic Differentiation (Autodiff, AD, etc.)

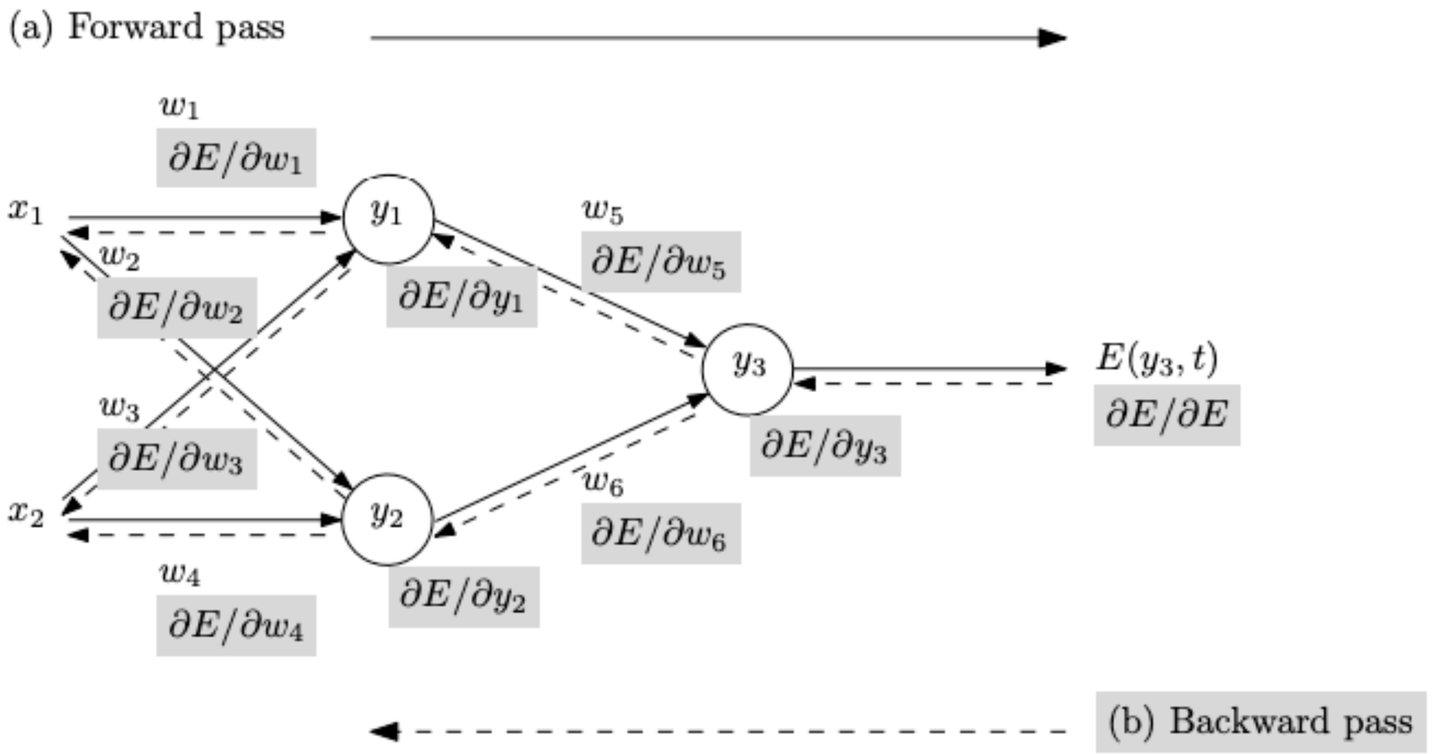
i.e., the chain-rule!



Now scale this to millions of operations!

Two Flavors of AD:

$$f(x_1, x_2) = E$$



Rule of Thumb:

Forwards: $f : R^n \to R^m, n \ll m$

Backwards: $f : R^n \to R^m, n \gg m$

Credit: https://gowrishankar.info

Where can I use AD in astronomy?

- Physics & Motion Simulations (e.g., N-body)
- Solving complex non-linear PDEs (e.g., Fluid Dynamics)
- Verification of code correctness through derivatives
- Information Theory (e.g., Fisher information of system)
- Probabilistic & Bayesian Modeling (see you next week)
- Complex, high-order, non-linear optimization & regression (AI / ML)

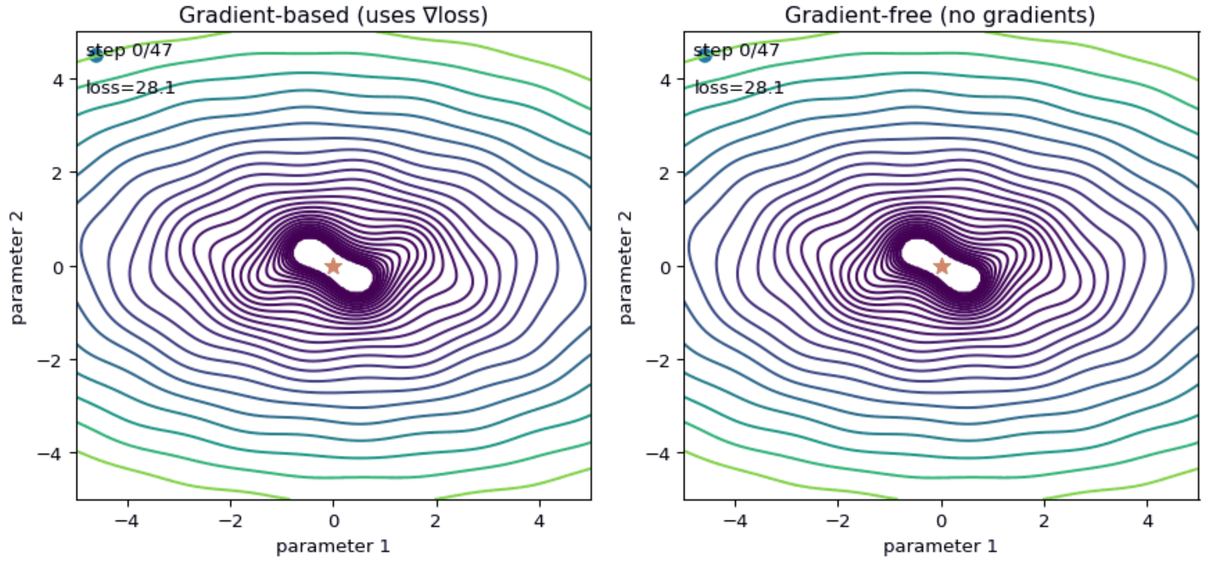
Example: Optimization of large models

Fit a complex model with lots of parameters to some data!

Loss Function: what you are trying to optimize in regression (e.g., mean-squared error)

Regression Approaches:

- Gradient-free Methods (e.g., Nelder-Mead, Simulated Annealing, LM)
- Gradient-based Methods (e.g., SGD, L-BFGS, Adam)



Example: Optimization of large models

Gradient-free Methods

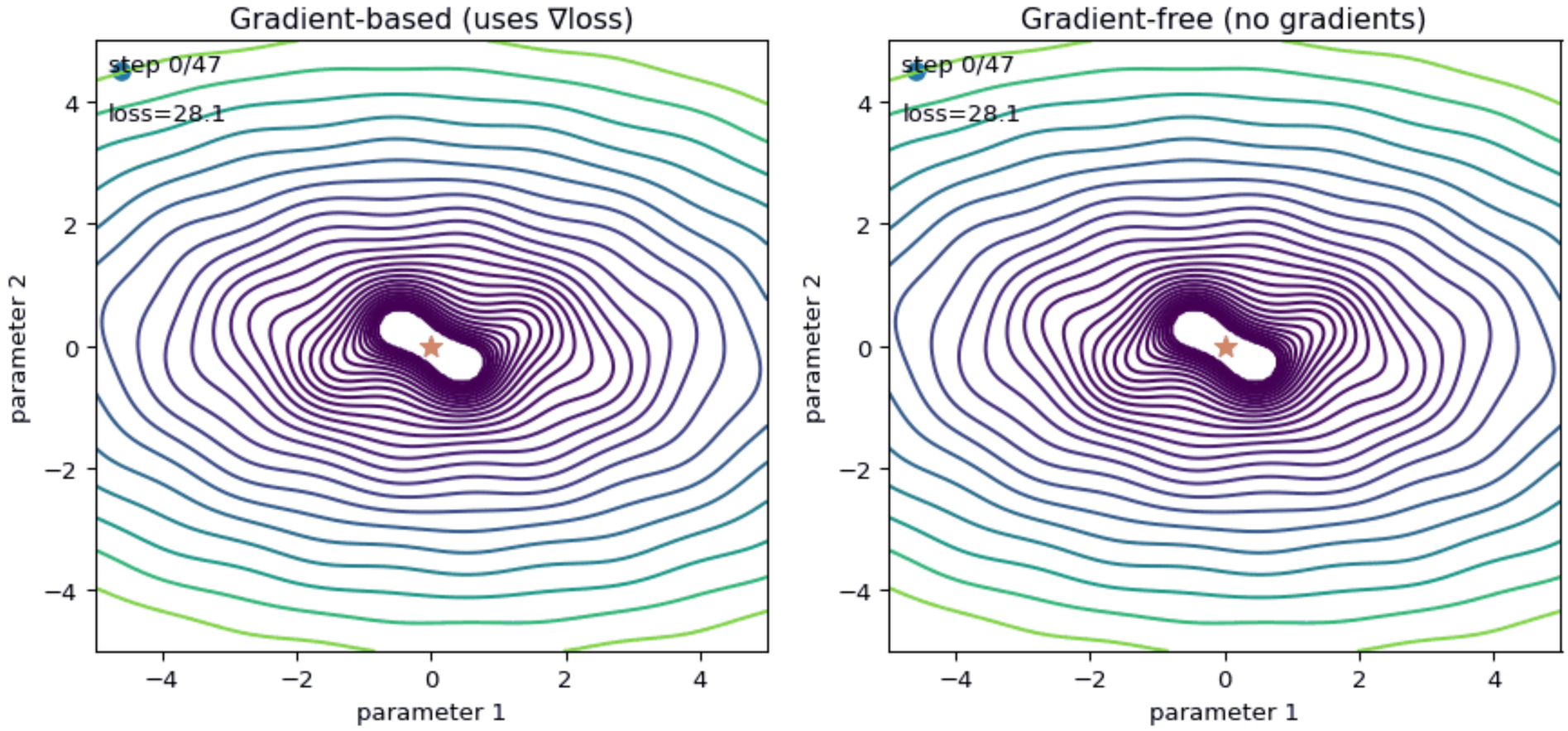
- Explores Loss Function without a priori knowledge to find optimal solution
- Optimization Scales ~ $O(N^2)$

Gradient-based Methods

- Uses partial derivatives of loss to decide how to navigate to the optimal solution
- Optimization Scales ~ O(N)

State-of-the-art LLM: 100's billions - trillions of free parameters

Any questions about AD?



`torch.autograd`

backwards pass

PyTorch Ecosystem

Model Building API

Flexible Training

`torch.autograd`
backwards pass

Dataloading

GPU Offloading