GT Deep Learning: 04

Vincent Perrollaz

Institut Denis Poisson

03 Mars 2026

Setup

Neural Network $\mathbf{x} \mapsto F(\mathbf{x}, p_1, \ldots, p_d)$

Supervised Learning Data $(\mathbf{x}_i, s_i)_{1 \leq i \leq N}$

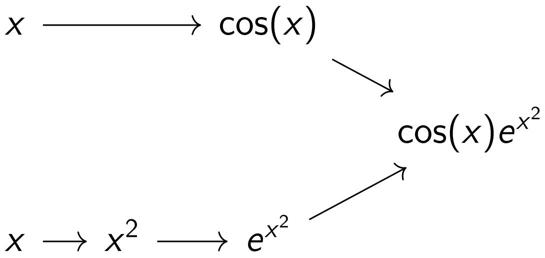
Maximum Likelihood Minimize $J(p_1, \ldots, p_d)$

Gradient Descent Compute $\nabla J(p_1, \ldots, p_d)$

Evaluation

- Evaluating an Expression (Meaning?)
- Parser: Expression $\mapsto$ Evaluation Tree
- ⚠ One Expression, Multiple Trees!
- Tree $\implies$ Recursive Structure
- Example: $u(x) = \cos(x)e^{x^2}$

Evaluation Tree



Evaluation: Left to Right with numbers!

Finite Difference

Formula $u'(x) \approx \frac{u(x+h)-u(x)}{h}$

Problem 1 Choice of h?

Problem 2 ⚠ Numerical Stability?

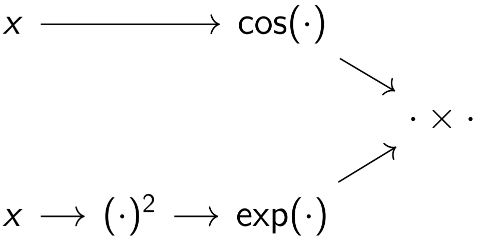
Problem 3 2 evaluation of u

Problem 3' ⚠ d variables $\Longrightarrow$ $d+1$ evaluations!

Symbolic Differenciation

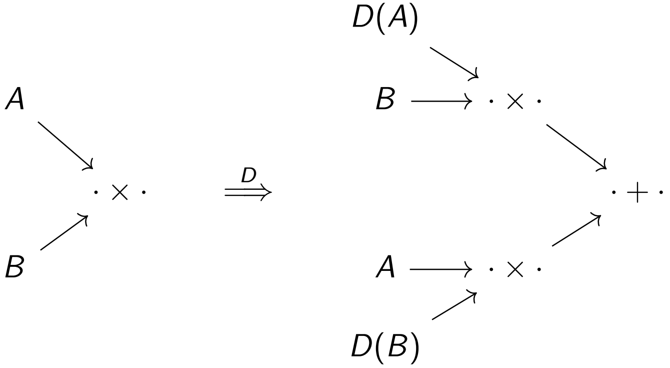
Goal $D: \text{ E.T.}(u) \mapsto \text{E.T.}(u')$

Reminder E.T. of $u(x) = \cos(x)e^{x^2}$:

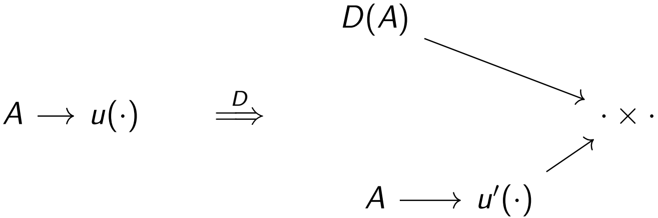


Multiplication Node

- Encodes $(uv)' = u'v + uv'$
- A and B subtrees



Composition Node



Summary

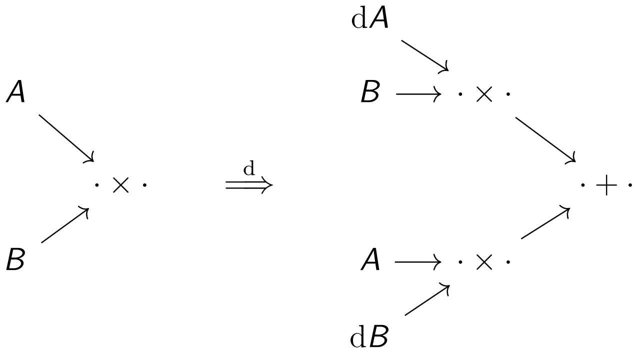
- From E.T.(u) Compute E.T.(u')!
- Problem 1: One tree for each $(\partial_{p_k})_{1 \leq k \leq d}$
- Problem 2: Combinatorial Explosion

Partial Derivative < Differential

- $F : \mathbb{R}^d \to \mathbb{R}$
- $\nabla F(\mathbf{p}) = (\partial_{p_1} F(\mathbf{p}), ..., \partial_{p_d} F(\mathbf{p}))$
- But also $\mathrm{d}F(\mathbf{p}) = \sum_{k=1}^{d} \partial_{p_k} F(\mathbf{p}) \mathrm{d}p_k$
- Compute $\mathrm{d}F$ E.T.

Multiplication Node II

- Encodes $\mathrm{d}(uv) = v\mathrm{d}u + u\mathrm{d}v$
- A and B subtrees



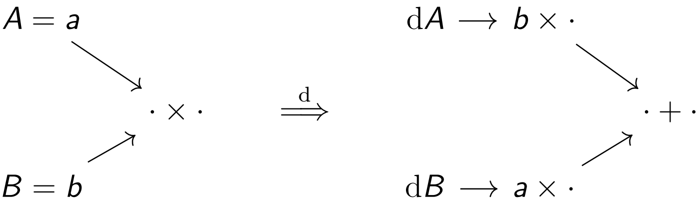
Main Idea of Automatic Differentiation

Forward Pass All nodes of E.T. have specific values

Backward Pass S.D. + Specific Values

Example Multiplication Node

Note $A\,B$: subtrees, $a\,b$: numbers



Example

- 1/2/1 Neural Network
- Explicit Formula
- ⚠ $\neq$ Parser
- Blackboard: Forward then Backward
- Demonstration using Pytorch