Learning advanced mathematical computations from examples

François Charton, Amaury Hayat, Guillaume Lample,

Facebook AI Research, Ecole des Ponts Paristech and Rutgers Camden

Is it possible to learn advanced math problems ?

...without any built-in mathematics

...using transformers

This requires to be accurate both in numerical and symbolic mathematical computations!

Can language models be used in maths?

Reasons to hope

- Mathematics is a language
- Datasets can be generated
- Solving a problem ⇔ translating it into its solutions

Reasons to doubt

- Can mathematics be learnt from examples?
- Can advanced problems be solved without rules and theory?

Three famous mathematical problems

- Local stability of differential systems
- Controllability of differential systems
- Existence and behavior of solutions of PDEs

All are a mixed setting involving symbolic and numerical computations.

Three famous mathematical problems

Local stability of differential systems:

Given a system x’ = f(x), find the speed of convergence to equilibrium.

$$\frac{dx_1}{dt} = \cos(x_2) - 1 - \sin(x_1)$$
$$\frac{dx_2}{dt} = x_1^2 - \sqrt{1 + x_2}$$

All solutions tends to 0 ?

Solved by the spectral mapping theorem

Three famous mathematical problems

Local controllability of differential systems:

Given a system x’ = f(x,u), find whether the system is locally controllable.

$$\frac{dx_1(t)}{dt} = \sin(x_1^2) + \log(1 + x_2) + \frac{\operatorname{atan}(ux_1)}{1 + x_2}$$
$$\frac{dx_2(t)}{dt} = x_2 - e^{x_1 x_2},$$

$\longrightarrow$

Can we reach any target x1 from any initial state x0 ?

Solved by Kalman’s rank criterion

Three famous mathematical problems

Existence and behavior of solutions of Partial Differential Equations:

given a linear PDE of the form $\partial_t u + D_x u = 0,$ and an initial condition u_0

$$D_x = 2\partial^2_{x_0} + 0.5\partial^2_{x_1} + \partial^4_{x_2} - 7\partial^2_{x_0,x_1} - 1.5\partial_{x_1}\partial^2_{x_2},$$

$$u_0(x) = e^{-3ix_2} x_0^{-1} \sin(x_0) e^{2.5ix_1} e^{-x_2^2},$$

$\longrightarrow$

Does a solution exist ?

Does it tend to 0 when time tends to infinity ?

Solved by Fourier transform on distributions

Let’s have a look at the computations involved !

Mathematical computations involved

Local controllability:

$$
\begin{aligned}
\frac{dx_1(t)}{dt} &= \sin(x_1^2) + \log(1 + x_2) + \frac{\operatorname{atan}(ux_1)}{1 + x_2} \\
\frac{dx_2(t)}{dt} &= x_2 - e^{x_1 x_2},
\end{aligned}
$$

- Differentiate with x

$$
A(x,u) = \begin{pmatrix} 2x_1 \cos(x_1^2) + \frac{u(1+x_2)^{-1}}{1+u^2x_1^2} & (1+x_2)^{-1} - \frac{\operatorname{atan}(ux_1)}{(1+x_2)^2} \\ -x_2 e^{x_1 x_2} & 1 - x_1 e^{x_1 x_2} \end{pmatrix}
$$

- Differentiate with u

$$
B(x,u) = \begin{pmatrix} x_1((1 + u^2 x_1^2)(1 + x_2))^{-1} \\ 0 \end{pmatrix}
$$

Mathematical computations involved

- Evaluate $$A(x_e, u_e) = \begin{pmatrix} 1.50 & 0.46 \\ -0.64 & 0.36 \end{pmatrix}, \quad B(x_e, u_e) = \begin{pmatrix} 0.27 \\ 0 \end{pmatrix}$$
- Compute control. matrix $$C = [B, AB]((x_e, u_e)) = \begin{pmatrix} 0.27 & 0.40 \\ 0 & -0.17 \end{pmatrix}$$
- Find the rank $$\mathbf{rank}(C) = 2$$
- Optionally compute a feedback matrix $$-B^{tr} \left(e^{-AT} \left[\int_0^T e^{-At} B B^{tr} e^{-A^{tr}t} dt \right] e^{-A^{tr}T} \right)^{-1}$$

Computations involved: linearization, differentiation, matrix product, integration, rank, etc.

Mathematical computations involved

$$\frac{dx_1(t)}{dt} = \sin(x_1^2) + \log(1 + x_2) + \frac{\mathrm{atan}(ux_1)}{1 + x_2}$$
$$\frac{dx_2(t)}{dt} = x_2 - e^{x_1 x_2},$$

$\longrightarrow$ Yes, locally controllable

Not quite straightforward to see !

How to represent mathematics as a natural language ?

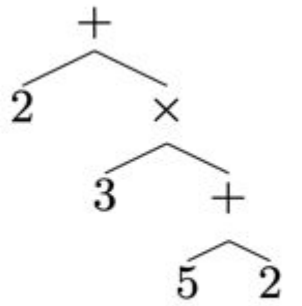
Mathematics as a natural language

Mathematical objects as sequences of tokens

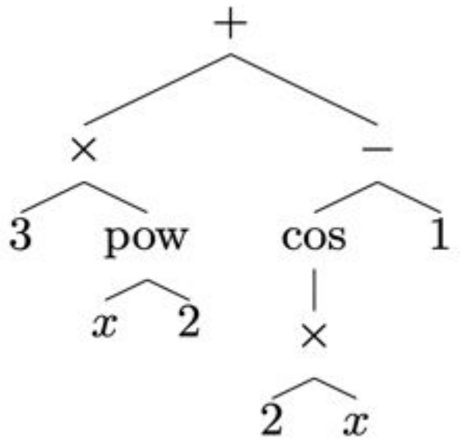
- Variables, functions and symbols are tokens
    - ‘sin’, ‘cos’, x’, ‘pi’
- Numbers can be represented in positional notation
    - 1214 = (‘+’, ‘1’, ‘2’, ‘1,’ ‘4’)
    - - 0.314 = (‘-’, ‘3’, ‘.’, ‘1’, ‘4’, ‘e’, ‘-’, ‘1’)
- Punctuation symbols take care of vectors and matrices
- Expressions (formulas, equations) as trees that can be enumerated as sequences of tokens

Expressions as trees

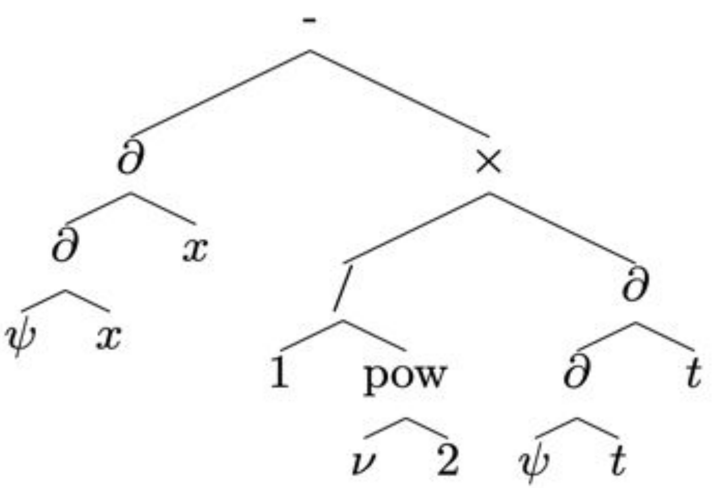
$2 + 3 \times (5 + 2)$



$3x^2 + \cos(2x) - 1$



$$\frac{\partial^2\psi}{\partial x^2} - \frac{1}{\nu^2}\frac{\partial^2\psi}{\partial t^2}$$

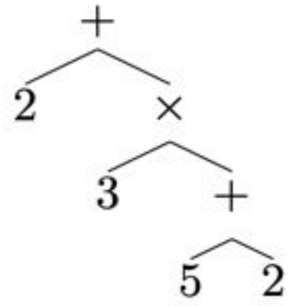


Trees as sequences

Preorder enumeration, aka normal Polish notation

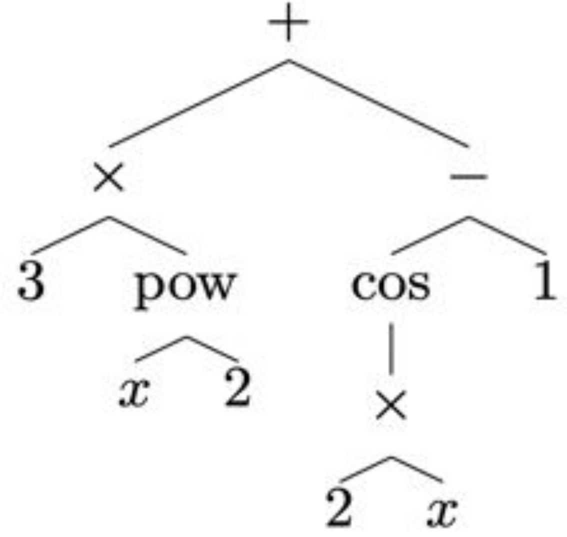
- begin from root
- parent before children
- left subtree before right subtree

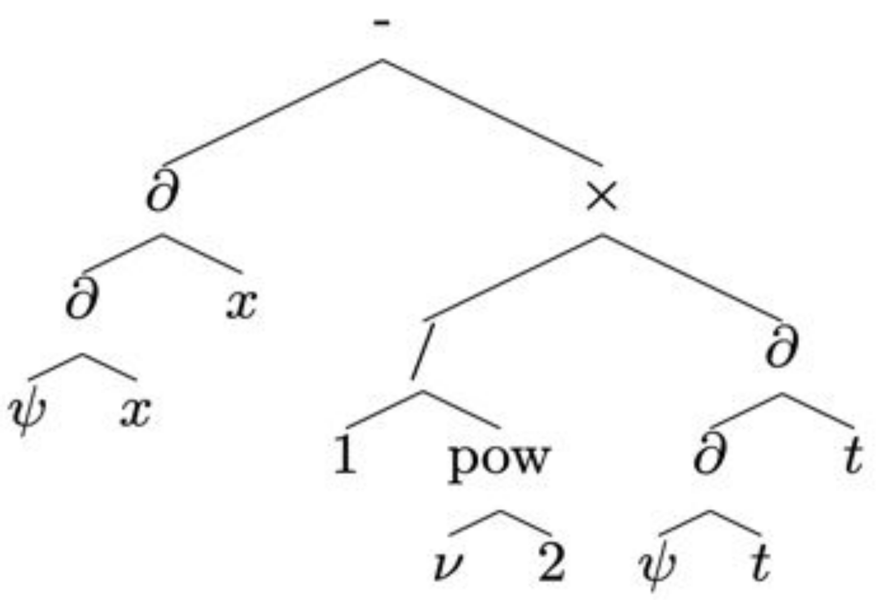
$$2 + 3 \times (5 + 2)$$



+ 2 * 3 + 5 2

Trees as sequences





+ * 3 pow x 2 - cos * 2 x 1

- ∂ ∂ ψ x x * / 1 pow ν 2 ∂ ∂ ψ t t

Generating datasets

Supervised learning : datasets of random problems and solutions

To generate a random problem

1. Generate a random tree
2. Randomly select operators as its internal nodes
3. Randomly select variables or constants as its leaves
4. Solve the problem using classical methods

Advantages: generation is iid, avoid bias.

Results

	6 layers dim. 512	1 layer dim. 512	FastText	Chance level
Stability	97.1	77.6	60.6	50
Controllability (3 to 6 eq.)	97.4	89.7	70.5	50
Non-autonomous control (2 to 3 eq.)	99.4	97.8	-	50
Linear PDE	98.6	96.4	-	75

High accuracy results, in all three tasks.

Generalization out of distribution

How does the trained model generalize ? when tested on a biased distribution

Table 8: **End to end stability: generalization over different test sets.**

	Overall	Degree 2	Degree 3	Degree 4	Degree 5
Baseline: training distribution	96.4	98.4	97.3	95.9	94.1
Unary operators: no trigs	95.7	98.8	97.3	95.5	91.2
Unary operators: no logs	95.3	98.2	97.1	95.2	90.8
Unary operators: no logs and trigs	95.7	98.8	97.7	95.2	91.0
Unary operators: less logs and trigs	95.9	98.8	96.8	95.0	93.1
Variables and integers: 10% integers	96.1	98.6	97.3	94.7	93.8
Variables and integers: 50% integers	95.6	97.8	96.7	94.3	93.1
Variables and integers: 70% integers	95.7	95.7	95.9	95.7	95.5

Does not change accuracy.

Generalization out of distribution

How easily can the model generate on larger sequence and larger problems?

$$\underbrace{\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{pmatrix}\sin(y) - x & x^2+1\\ xy+4 & \tan(y)\end{pmatrix}}_{\text{training data}} \quad \underbrace{\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix} = \begin{pmatrix}e^z\cos(x) & 2y-x & z^2+\sin(y)\\ \frac{y}{2}+4 & \frac{\tan(z)}{\ln(2+y)} & xz\\ \sqrt{x^2z+1} & \ln(1+y) & \sin(x^2)\end{pmatrix}}_{\text{testing data}}$$

	Overall	Degree 2	Degree 3	Degree 4	Degree 5
Baseline: training distribution	96.4	98.4	97.3	95.9	94.1
Expression lengths: n+3 to 3n+3	89.5	96.5	92.6	90.0	77.9
Expression lengths: 2n+3 to 4n+3	79.3	93.3	88.3	73.4	58.2
System degree: degree 6	78.7				

Thank you for watching !

Want to become a Transmathematican too ? Code and datasets are available:

https://github.com/facebookresearch/MathsFromExamples