Deep Learning for Symbolic Mathematics

Guillaume Lample
Facebook AI Research
glample@fb.com

François Charton
Facebook AI Research
fcharton@fb.com

Recordatorio breve del artículo original

- Se considera resolver dos problemas de matemáticas simbólicas: **integración de funciones** y **solución de ecuaciones diferenciales ordinarias** de primer y segundo orden.
- Se propone utilizar un **modelo transformer** para atacar ambos problemas.

Functions and their primitives generated with the forward approach (FWD)	
$\cos^{-1}(x)$	$x\cos^{-1}(x) - \sqrt{1-x^2}$
$x\,(2x + \cos(2x))$	$\frac{2x^3}{3} + \frac{x\sin(2x)}{2} + \frac{\cos(2x)}{4}$
$\frac{x\,(x+4)}{x+2}$	$\frac{x^2}{2} + 2x - 4\log(x+2)$
$\frac{\cos(2x)}{\sin(x)}$	$\frac{\log(\cos(x)-1)}{2} - \frac{\log(\cos(x)+1)}{2} + 2\cos(x)$
$3x^2\sinh^{-1}(2x)$	$x^3\sinh^{-1}(2x) - \frac{x^2\sqrt{4x^2+1}}{6} + \frac{\sqrt{4x^2+1}}{12}$
$x^3\log\left(x^2\right)^4$	$\frac{x^4\log\left(x^2\right)^4}{4} - \frac{x^4\log\left(x^2\right)^3}{2} + \frac{3x^4\log\left(x^2\right)^2}{4} - \frac{3x^4\log\left(x^2\right)}{4} + \frac{3x^4}{8}$

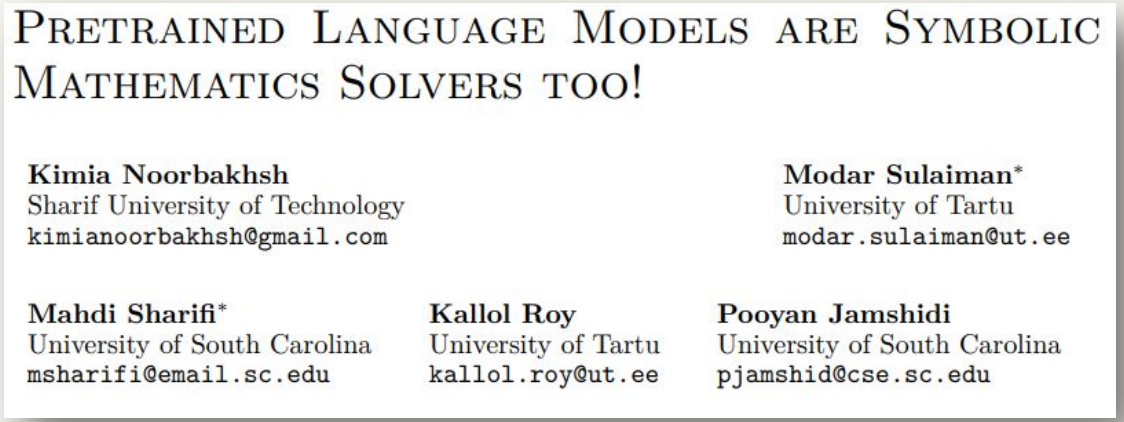
Tabla 9 - Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics*.

Functions and their primitives generated with the backward approach (BWD)	
$\cos(x) + \tan^2(x) + 2$	$x + \sin(x) + \tan(x)$
$\frac{1}{x^2\sqrt{x-1}\sqrt{x+1}}$	$\frac{\sqrt{x-1}\sqrt{x+1}}{x}$
$\left(\frac{2x}{\cos^2(x)} + \tan(x)\right)\tan(x)$	$x\tan^2(x)$
$\frac{x\tan\left(\frac{e^x}{x}\right) + \frac{(x-1)e^x}{\cos^2\left(\frac{e^x}{x}\right)}}{x}$	$x\tan\left(\frac{e^x}{x}\right)$
$1 + \frac{1}{\log(\log(x))} - \frac{1}{\log(x)\log(\log(x))^2}$	$x + \frac{x}{\log(\log(x))}$
$-2x^2\sin\left(x^2\right)\tan(x) + x\left(\tan^2(x) + 1\right)\cos\left(x^2\right) + \cos\left(x^2\right)\tan(x)$	$x\cos\left(x^2\right)\tan(x)$

Tabla 9 - Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics*.

Propuesta y justificación

- Propuesta: Realizar un fine-tuning de alguno de los modelos pre-entrenados para emplearlo en otra tarea similar.
  - Existen algunos trabajos que respaldan el fine-tuning en modelos de arquitectura tipo transformers para este tipo de problemas, por ejemplo:



PRETRAINED LANGUAGE MODELS ARE SYMBOLIC MATHEMATICS SOLVERS TOO!

Kimia Noorbakhsh
Sharif University of Technology
kimianoorbakhsh@gmail.com

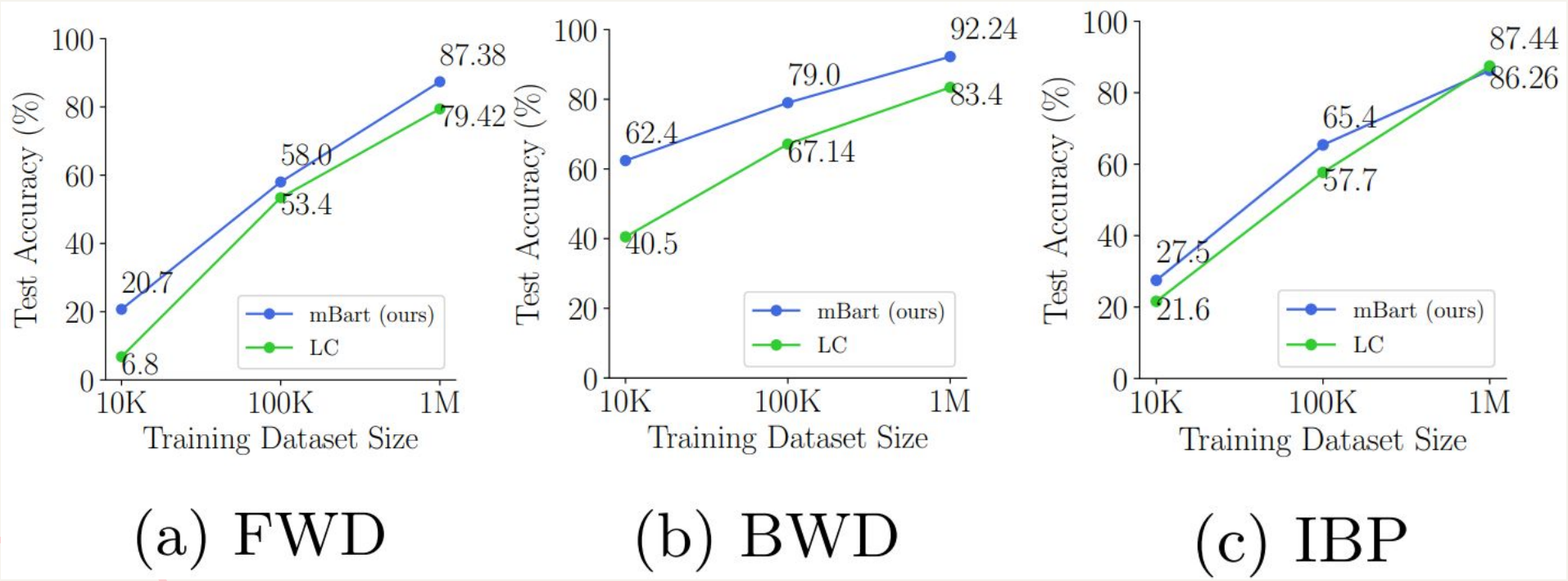
Modar Sulaiman*
University of Tartu
modar.sulaiman@ut.ee

Mahdi Sharifi*
University of South Carolina
msharifi@email.sc.edu

Kallol Roy
University of Tartu
kallol.roy@ut.ee

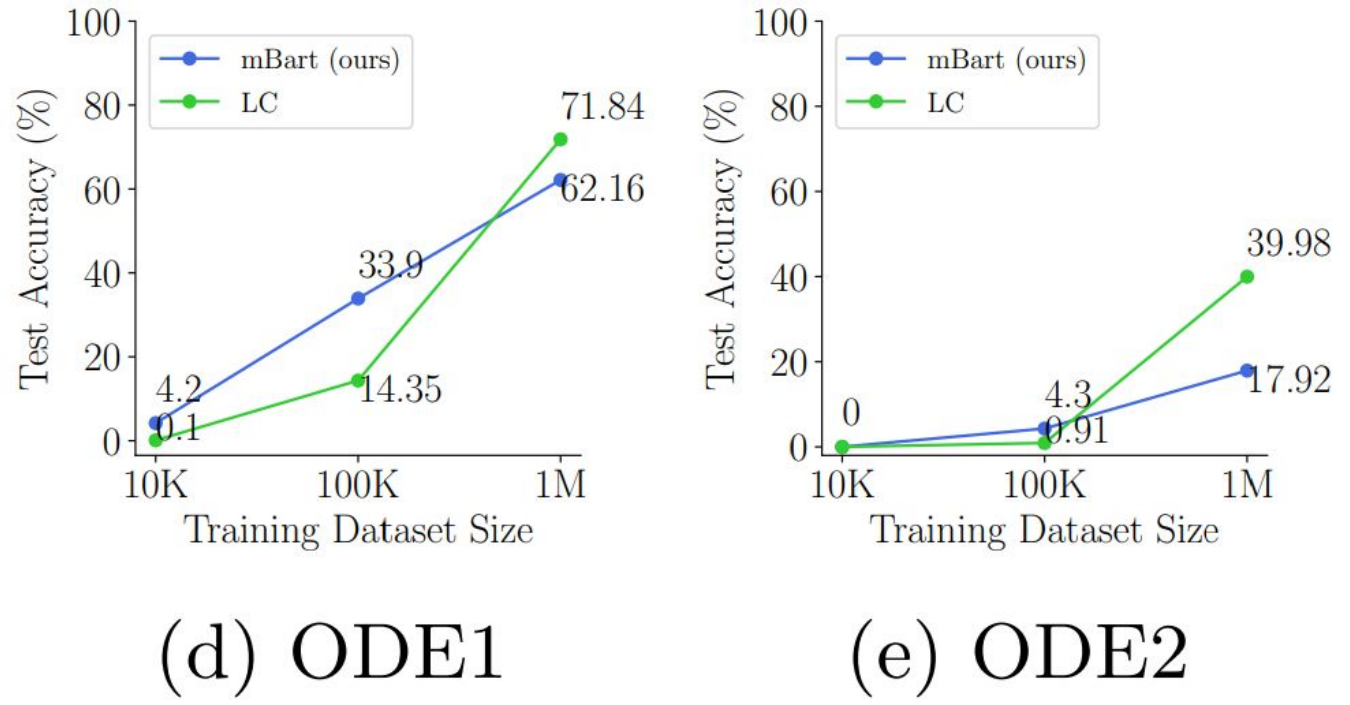
Pooyan Jamshidi
University of South Carolina
pjamshid@cse.sc.edu

Propuesta y justificación



Noorbakhsh, K., Sulaiman, M., Sharifi, M., Roy, K., & Jamshidi, P. (2021). *Pretrained Language Models are Symbolic Mathematics Solvers too!*

Propuesta y justificación



Noorbakhsh, K., Sulaiman, M., Sharifi, M., Roy, K., & Jamshidi, P. (2021). *Pretrained Language Models are Symbolic Mathematics Solvers too!*

Propuesta y justificación

- Se toman los datos de las integrales generadas por el método forward.

$$\int \frac{x(x+4)}{x+2}\,dx \qquad\qquad \frac{x^2}{2} + 2x - 4\log(x+2) + C$$

- Se creará una nueva tarea, cálculo de derivadas. La ecuación de entrada ahora corresponde a la ecuación objetivo, y la ecuación objetivo ahora es la ecuación de entrada.

$$f = \frac{x^2}{2} + 2x - 4\log(x+2) \qquad\qquad f' = \frac{x(x+4)}{x+2}\,dx$$

- Se empleará el modelo para cálculo de integrales pre-entrenado con fórmulas generadas con el método backward.

Metodología de evaluación

Se realizarán 3 experimentos con los datos para cálculo de derivadas.

- Para los primeros dos, se tomará el modelo bwd y se entrenará con 100K y 500K fórmulas cada uno.

- El tercer modelo tomará de base el modelo fwd y se entrenará con 100K.

Todos los modelos se evaluarán con 2K datos de validación y prueba, y un beam size de 1 con la métrica de exactitud expresada en porcentaje.

$$exactitud = \frac{VP + VN}{VP + FP + FN + VN} * 100$$

Arquitecturas y/o hiperparámetros

Permanece la misma arquitectura e hiperparámetros, a excepción del tamaño del lote que cambia a 128:

- 8 Cabezas de atención (encoder y decoder).
- 6 Capas/Bloques transformers para el encoder y 6 para el decoder.
- Dimensionalidad del embedding de 1024.
- Optimizador Adam con una tasa de aprendizaje de 10^{-4}.
- 256 ecuaciones por lote -> 128 ecuaciones por lote.

Resultados

	Entrenamiento con el modelo bwd 100K entrenamiento	Entrenamiento con el modelo bwd 500K entrenamiento	Entrenamiento con el modelo fwd 100K entrenamiento
Época 1	Validación: 66.16% Prueba: 66.56%	Validación: 70.41% Prueba: 70.26%	Validación: 58.37% Prueba: 58.57%
Época 2	**Validación: 67.96%** **Prueba: 66.61%**	Validación: 70.81% Prueba: 70.91%	Validación: 59.57% Prueba: 59.97%
Época 3	Validación: 67.06% Prueba: 65.77%	**Validación: 70.96%** **Prueba: 71.71%**	**Validación: 61.12%** **Prueba: 60.82%**

Tamaño del conjunto de validación: 2K fórmulas
Tamaño del conjunto de prueba: 2K fórmulas

Discusión

- La transferencia de conocimiento hacía problemas similares, en arquitecturas tipo transformer, **ayudó al modelo a entrenarse con menos épocas** a pesar de tener pocos datos.

- El modelo alcanza un resultado bueno con 500K datos para el conjunto de prueba (71.7%) en 3 épocas, sin embargo, debido a la arquitectura y naturaleza del problema, **se requiere de más datos** si se desea alcanzar algo cercano al 90% para un beam size de 1.

Discusión

Training data	Forward (FWD)			Backward (BWD)			Integration by parts (IBP)		
	Beam 1	Beam 10	Beam 50	Beam 1	Beam 10	Beam 50	Beam 1	Beam 10	Beam 50
FWD	93.6	95.6	96.2	10.9	13.9	17.2	85.6	86.8	88.9
BWD	18.9	24.6	27.5	98.4	99.4	99.7	42.9	54.6	59.2
BWD + IBP	41.6	54.9	56.1	98.2	99.4	99.7	96.8	99.2	99.5
BWD + IBP + FWD	89.1	93.4	94.3	98.1	99.3	99.7	97.2	99.4	99.7

Tabla 6 - Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics*.

Discusión

Linear algebra with transformers

François Charton Meta AI
fcharton@meta.com

Transactions in Machine Learning Research, October 2022

Referencias

Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics*. http://arxiv.org/abs/1912.01412

Noorbakhsh, K., Sulaiman, M., Sharifi, M., Roy, K., & Jamshidi, P. (2021). *Pretrained Language Models are Symbolic Mathematics Solvers too!* http://arxiv.org/abs/2110.03501

Charton, F. (2021). *Linear algebra with transformers*. http://arxiv.org/abs/2112.01898

FIN

Introducción

- Los problemas de **razonamiento simbólico** con redes neuronales resultaban desafiantes. Inicialmente, se empleaban modelos de **redes neuronales recursivas/Tree-LSTM**, ya que las expresiones fácilmente se pueden mapear a estructuras de tipo árbol.

- Por ejemplo, para decidir si una igualdad matemática es válida dado su árbol sintáctico.

- Por otro lado, se han probado modelos de **redes neuronales recurrentes y transformers**, para el caso de manipulación de operaciones aritméticas, solución de sistemas de ecuaciones lineales de 1 y 2 variables, ordenamiento de números y derivadas de polinomios. Sin embargo, no existía algún trabajo con razonamiento simbólico más complejo.

Propuesta

- En este trabajo se considera resolver dos problemas de matemáticas simbólicas: **integración de funciones** y **solución de ecuaciones diferenciales ordinarias** de primer y segundo orden.

- Se propone utilizar un **modelo transformer** para atacar ambos problemas.

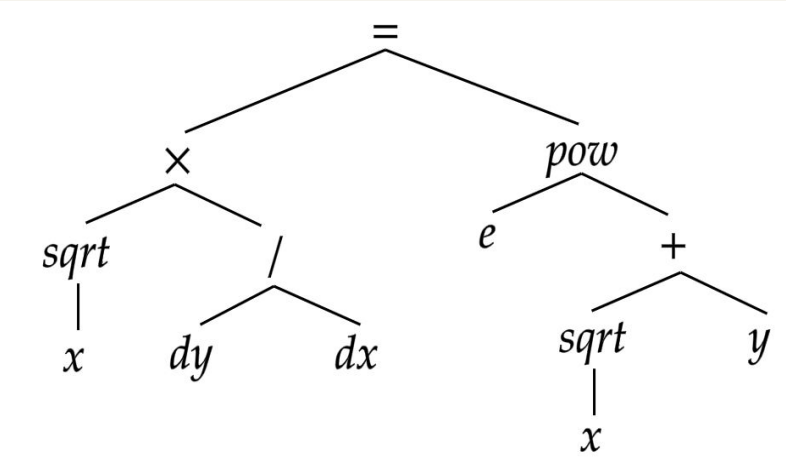
$$\int 3x^2 + cos(2x) - 1 \; dx$$

$$\sqrt{x}\frac{dy}{dx} = e^{\sqrt{x+y}}$$

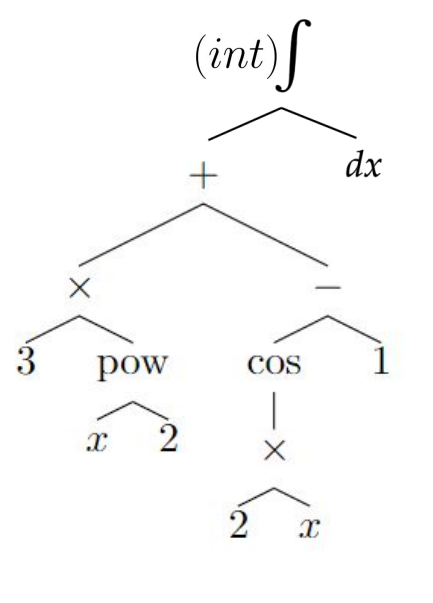
Los **árboles sintácticos** eliminan la necesidad de preocuparse por la **precedencia, la asociatividad y el uso de paréntesis**.

Pero... ¿Cómo **representar los árboles sintácticos** de las expresiones matemáticas como una **secuencia**?

$$\sqrt{x}\frac{dy}{dx} = e^{\sqrt{x+y}}$$



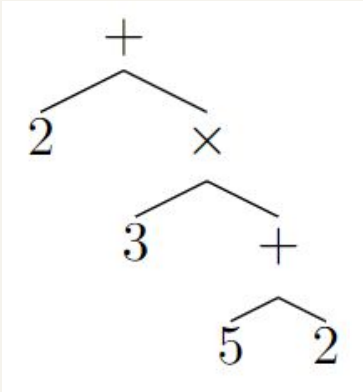
$$\int 3x^2 + cos(2x) - 1 \; dx$$



Expresión matemática

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

Árbol sintáctico



Notación prefija

$[+2 \times 3 + 5\ 2]$

Generación del conjunto de datos

Las funciones generadas aleatoriamente consideran lo siguiente:

- Árboles con a lo más **n=15** nodos internos (operadores).
- **Operadores unarios**
  $(exp, log, sqrt, sin, cos, tan, sin^{-1}, cos^{-1}, tan^{-1}, sinh, cosh, tanh)$
- **Operadores binarios** (+, - , x , /)
- Las hojas contienen **variables** $\{x\}$, **constantes numéricas** {-5,...,5} \ {0} **y simbólicas** $\{e, \pi\}$.

Para el conjunto de datos además se requiere los **problemas** y sus **soluciones**.

- Función y su integral.

Integrales

Forward generation (FWD): Construir una **función aleatoria** de n operadores y **calcular sus integrales** con algún sistema algebraico computacional. Las funciones que no se pueden integrar se descartan.

- Representa el subconjunto de problemas que pueden ser solucionados por un programa externo.
- Expresiones largas demoran en calcularse.

Backward generation (BWD): Construir una **función aleatoria f**, **calcular su derivada f'** y agregar el par (f', f) al conjunto de datos.

- La derivada siempre es posible y es más rápida para expresiones grandes.

Backward generation with integration by parts (IBP): Dadas dos funciones f y g generadas aleatoriamente, calcular sus respectivas derivadas f', g'. Si f'g o fg' ya está en el conjunto de datos, se conoce su integral y se puede calcular la integral como: $\int \boldsymbol{f g'} = \boldsymbol{fg} - \int \boldsymbol{f' g}$. Si no se genera otra función f y g aleatoriamente.

- Funciones simples con integrales complejas tienen baja probabilidad de generarse, con esto se soluciona este problema.

Functions and their primitives generated with the forward approach (FWD)	
$\cos^{-1}(x)$	$x\cos^{-1}(x) - \sqrt{1-x^2}$
$x\left(2x + \cos(2x)\right)$	$\frac{2x^3}{3} + \frac{x\sin(2x)}{2} + \frac{\cos(2x)}{4}$
$\frac{x(x+4)}{x+2}$	$\frac{x^2}{2} + 2x - 4\log(x+2)$
$\frac{\cos(2x)}{\sin(x)}$	$\frac{\log(\cos(x)-1)}{2} - \frac{\log(\cos(x)+1)}{2} + 2\cos(x)$
$3x^2\sinh^{-1}(2x)$	$x^3\sinh^{-1}(2x) - \frac{x^2\sqrt{4x^2+1}}{6} + \frac{\sqrt{4x^2+1}}{12}$
$x^3\log\left(x^2\right)^4$	$\frac{x^4\log\left(x^2\right)^4}{4} - \frac{x^4\log\left(x^2\right)^3}{2} + \frac{3x^4\log\left(x^2\right)^2}{4} - \frac{3x^4\log\left(x^2\right)}{4} + \frac{3x^4}{8}$

Functions and their primitives generated with the backward approach (BWD)	
$\cos(x) + \tan^2(x) + 2$	$x + \sin(x) + \tan(x)$
$\dfrac{1}{x^2\sqrt{x-1}\sqrt{x+1}}$	$\dfrac{\sqrt{x-1}\sqrt{x+1}}{x}$
$\left(\dfrac{2x}{\cos^2(x)} + \tan(x)\right)\tan(x)$	$x\tan^2(x)$
$\dfrac{x\tan\left(\frac{e^x}{x}\right) + \frac{(x-1)e^x}{\cos^2\left(\frac{e^x}{x}\right)}}{x}$	$x\tan\left(\dfrac{e^x}{x}\right)$
$1 + \dfrac{1}{\log(\log(x))} - \dfrac{1}{\log(x)\log(\log(x))^2}$	$x + \dfrac{x}{\log(\log(x))}$
$-2x^2\sin\left(x^2\right)\tan(x) + x\left(\tan^2(x)+1\right)\cos\left(x^2\right) + \cos\left(x^2\right)\tan(x)$	$x\cos\left(x^2\right)\tan(x)$

Functions and their primitives generated with the integration by parts approach (IBP)	
$x\left(x+\log\left(x\right)\right)$	$\frac{x^2\left(4x+6\log\left(x\right)-3\right)}{12}$
$\frac{x}{\left(x+3\right)^2}$	$\frac{-x+\left(x+3\right)\log\left(x+3\right)}{x+3}$
$\frac{x+\sqrt{2}}{\cos^2\left(x\right)}$	$\left(x+\sqrt{2}\right)\tan\left(x\right)+\log\left(\cos\left(x\right)\right)$
$x\left(2x+5\right)\left(3x+2\log\left(x\right)+1\right)$	$\frac{x^2\left(27x^2+24x\log\left(x\right)+94x+90\log\left(x\right)\right)}{18}$
$\frac{\left(x-\frac{2x}{\sin^2\left(x\right)}+\frac{1}{\tan\left(x\right)}\right)\log\left(x\right)}{\sin\left(x\right)}$	$\frac{x\log\left(x\right)+\tan\left(x\right)}{\sin\left(x\right)\tan\left(x\right)}$
$x^3\sinh\left(x\right)$	$x^3\cosh\left(x\right)-3x^2\sinh\left(x\right)+6x\cosh\left(x\right)-6\sinh\left(x\right)$

Ecuaciones diferenciales ordinarias

Generate a random function	$f(x) = c_1 e^x + c_2 e^{-x}$
Solve in c_2	$c_2 = f(x)e^x - c_1 e^{2x} = F(x, f(x), c_1)$
Differentiate in x	$e^x \left(f'(x) + f(x)\right) - 2c_1 e^{2x} = 0$
Solve in c_1	$c_1 = \frac{1}{2} e^{-x} \left(f'(x) + f(x)\right) = G(x, f(x), f'(x))$
Differentiate in x	$0 = \frac{1}{2} e^{-x} \left(f''(x) - f(x)\right)$
Simplify	$y'' - y = 0$

Transformaciones al conjunto de datos

- **Simplificación de expresiones.**
  Ejemplo: $x + 1 + 3 + 5 * 2 = x + 14$ $\log\left(e^{x+3}\right) = x + 3$

- **Simplificación de constantes.**
  Ejemplo: $x + \boldsymbol{x\,tan(3)} + \boldsymbol{xc} + 1 = x + \boldsymbol{xc} + 1$

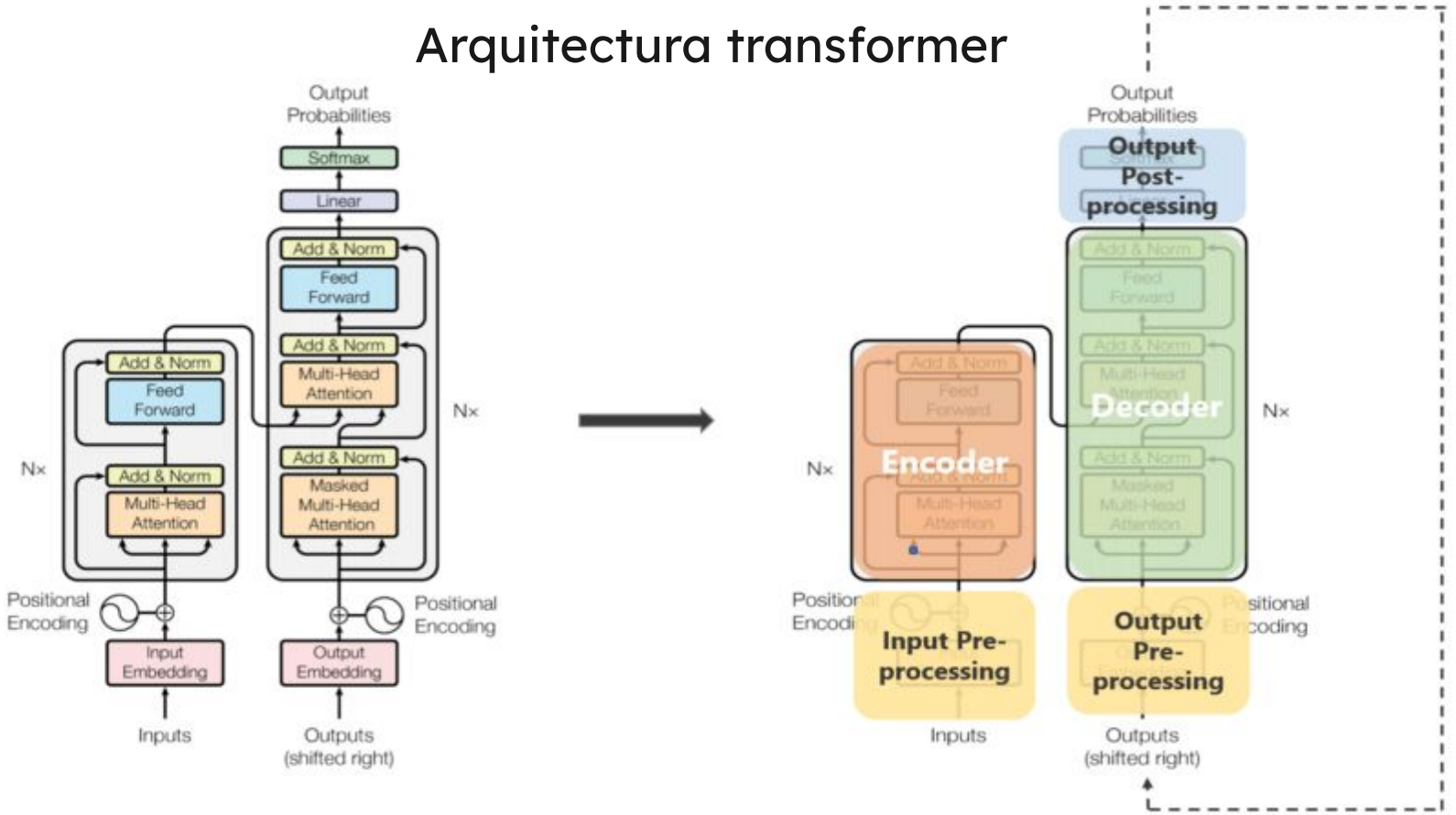
- **Eliminación de expresiones con valores inválidos.**
  Ejemplo: $\log(0)$ $\sqrt{-2}$

Conjunto de datos

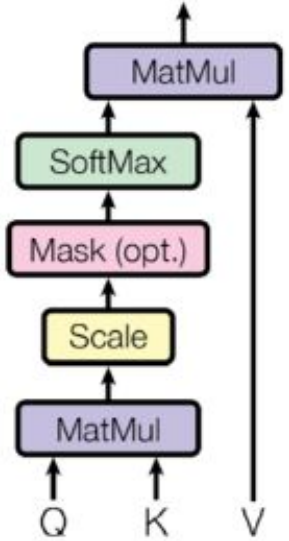
	Forward	Backward	Integration by parts	ODE 1	ODE 2
Training set size	20M	40M	20M	40M	40M
Input length	18.9±6.9	70.2±47.8	17.5±9.1	123.6±115.7	149.1±130.2
Output length	49.6±48.3	21.3±8.3	26.4±11.3	23.0±15.2	24.3±14.9
Length ratio	2.7	0.4	2.0	0.4	0.1
Input max length	69	450	226	508	508
Output max length	508	75	206	474	335

Table 1: **Training set sizes and length of expressions (in tokens) for different datasets.** FWD and IBP tend to generate examples with outputs much longer than the inputs, while the BWD approach generates shorter outputs. Like in the BWD case, ODE generators tend to produce solutions much shorter than their equations.

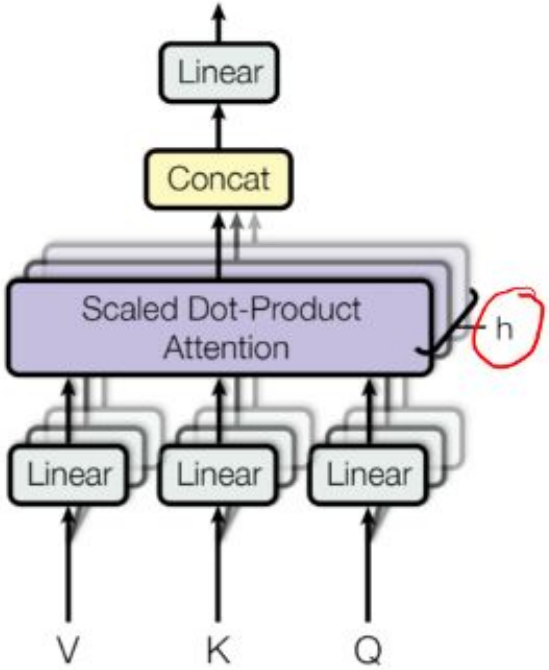
Arquitectura transformer



Scaled Dot-Product Attention



Multi-Head Attention



- Se entrenó un modelo transformer con las siguientes características:

  - 8 Cabezas de atención.
  - 6 Capas.
  - Dimensionalidad de 512.
  - Optimizador Adam con una tasa de aprendizaje de 10^{-4}.

Beam search

$$162x\log(x)y' + 2y^3\log(x)^2 - 81y\log(x) + 81y = 0 \qquad y = \frac{9\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{\sqrt{c+2x}}$$

Hypothesis	Score	Hypothesis	Score
$\frac{9\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{\sqrt{c+2x}}$	-0.047	$\frac{9}{\sqrt{\frac{c\log(x)}{x} + 2\log(x)}}$	-0.124
$\frac{9\sqrt{x}}{\sqrt{c+2x}\sqrt{\log(x)}}$	-0.056	$\frac{9\sqrt{x}}{\sqrt{c\log(x) + 2x\log(x)}}$	-0.139
$\frac{9\sqrt{2}\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{2\sqrt{c+x}}$	-0.115	$\frac{9}{\sqrt{\frac{c}{x}+2}\sqrt{\log(x)}}$	-0.144
$9\sqrt{x}\sqrt{\frac{1}{c\log(x) + 2x\log(x)}}$	-0.117	$9\sqrt{\frac{1}{\frac{c\log(x)}{x} + 2\log(x)}}$	-0.205
$\frac{9\sqrt{2}\sqrt{x}}{2\sqrt{c+x}\sqrt{\log(x)}}$	-0.124	$9\sqrt{x}\sqrt{\frac{1}{c\log(x) + 2x\log(x) + \log(x)}}$	-0.232

Table 5: Top 10 generations of our model for the first order differential equation $162x\log(x)y' + 2y^3\log(x)^2 - 81y\log(x) + 81y = 0$, generated with a beam search. All hypotheses are valid solutions, and are equivalent up to a change of the variable c. Scores are log-probabilities normalized by sequence lengths.

- Existen múltiples soluciones que son equivalentes, pero escritas de otra manera.
- El decodificador no está exento de generar expresiones infijas inválidas. Se deben utilizan restricciones. En este caso solo se ignoran

Beam search

```
Input function f: x*(2*exp(x)*cos(x + exp(x))*cos(x**2 + 2) - (exp(x) + 1)*exp(x)*sin(x + exp(x))*sin(x**2 + 2)/x + exp
(x)*sin(x**2 + 2)*cos(x + exp(x))/x - exp(x)*sin(x**2 + 2)*cos(x + exp(x))/x**2)*exp(-x)/(sin(x**2 + 2)*cos(x + exp
(x)))
Reference function F: log(exp(x)*sin(x**2 + 2)*cos(x + exp(x))/x)

-0.00003  OK  log(exp(x)*sin(x**2 + 2)*cos(x + exp(x))/x)
-0.28475  OK  log(exp(x)*sin((x**3 + 2*x)/x)*cos(x + exp(x))/x)
-0.28592  OK  log(exp(x)*sin(x*(x + 2/x))*cos(x + exp(x))/x)
-0.35794  OK  log(exp(x)*sin(x*(x + 1) - x + 2)*cos(x + exp(x))/x)
-0.37952  NO  log(exp(x)*sin(x**2*(x + 2/x))*cos(x + exp(x))/x)
-0.38034  NO  log(exp(x)*sin(x**2 + 2)*cos(x + sinh(x) + cosh(x))/x)
-0.39518  OK  atan(tan(log(exp(x)*sin(x**2 + 2)*cos(x + exp(x))/x)))
-0.39689  OK  log(exp(x)*sin(x*(x - 1) + x + 2)*cos(x + exp(x))/x)
-0.43203  NO  log(exp(x)*sin((x**2 + 2)**2)*cos(x + exp(x))/x)
-0.44538  NO  log(exp(x)*sin(x**2 + 2*x)*cos(x + exp(x))/x)
```

- Existen soluciones válidas y no válidas.

Resultados

- Se empleo SymPy para simplificar y comparar si son iguales.
- En el caso de ecuaciones diferenciales basta con reemplazar la solución en la ecuación y al simplificar verificar que se cumpla la igualdad.

	Integration (FWD)	Integration (BWD)	Integration (IBP)	ODE (order 1)	ODE (order 2)
Beam size 1	93.6	98.4	96.8	77.6	43.0
Beam size 10	95.6	99.4	99.2	90.5	73.0
Beam size 50	96.2	99.7	99.5	94.0	81.2

Table 2: **Accuracy of our models on integration and differential equation solving.** Results are reported on a held out test set of 5000 equations. For differential equations, using beam search decoding significantly improves the accuracy of the model.

- Se utilizó la métrica de exactitud.

$$exactitud = \frac{VP + VN}{VP + FP + FN + VN}$$

Resultados

	Integration (BWD)	ODE (order 1)	ODE (order 2)
Mathematica (30s)	84.0	77.2	61.6
Matlab	65.2	-	-
Maple	67.4	-	-
Beam size 1	98.4	81.2	40.8
Beam size 10	99.6	94.0	73.2
Beam size 50	99.6	97.0	81.0

Table 3: **Comparison of our model with Mathematica, Maple and Matlab on a test set of 500 equations.**

Timeout para los sistemas algebraicos de cómputo

Timeout (s)	Success	Failure	Timeout
5	77.8	9.8	12.4
10	82.2	11.6	6.2
30	84.0	12.8	3.2
60	84.4	13.4	2.2
180	84.6	13.8	1.6

Table 8: **Accuracy of Mathematica on 500 functions to integrate, for different timeout values.** As the timeout delay increases, the percentage of failures due to timeouts decreases. With a limit of 3 minutes, timeouts only represent 10% of failures. As a result, the accuracy without timeout would not exceed 86.2%.

Conclusiones

- **Modelos transformers** pueden ser aplicados a tareas difíciles de matemáticas simbólicas, como integración de funciones y solución de ecuaciones diferenciales ordinarias.
- Este modelo **se comportó mejor** para este tipo de problemas en cuanto a encontrar una solución, **comparado con los sistemas algebraicos computacionales** actuales que dependen de algoritmos y heurísticas complejas, por ejemplo, el algoritmo de Risch.
- El modelo es capaz de **generar soluciones equivalentes**.
- Es necesario considerar **un valor mayor de ancho (beam width) en el beam search** para algunos problemas como lo fue **ecuaciones diferenciales**.
- Para generar el **conjunto de datos** en este caso se **depende de un sistema externo** capaz de integrar y de derivar.

FIN