Do Bag of Words ao Transformer: A Evolução do NLP

Luiz Nonenmacher
Marcelo Prates

THE DEVELOPER'S CONFERENCE

poatek

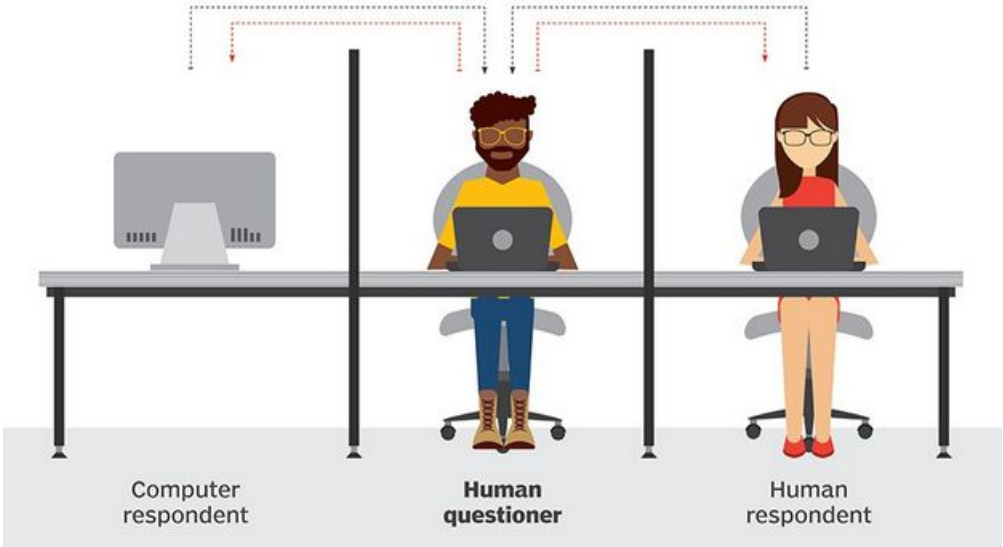
Outline

1. Natural Language Processing
2. One-hot encoding
3. Bag-of-words / TF-IDF
4. Word embeddings
5. Recurrent Neural Networks
6. Attention
7. Transformer
8. Aplicações

Natural Language Processing

- Processamento de língua natural (NLP) é uma subárea da ciência da computação, inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de línguas humanas naturais.

- Uma das áreas mais tradicionais dentro de AI e mais relacionada com o que pensamos como “inteligência” (ex: Teste de Turing).



Natural Language Processing

The Wolf of Wall Street.

One Star

By Joe Watson - December 14, 2014

There were no wolves in the movie.

0 of 3 people found this review helpful

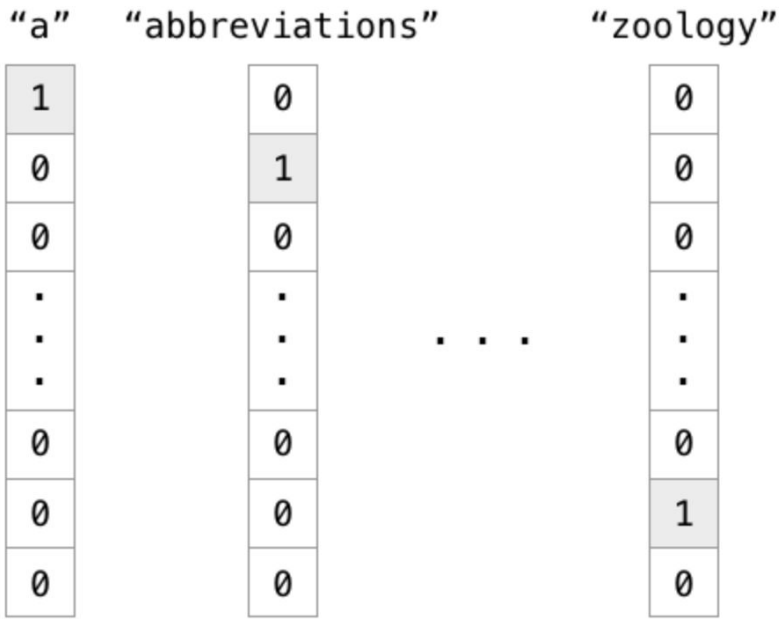
Back in 2000 , People Magazine PUBLISHER highlighted Prince Williams' PERSON style who at the time was a little more fashion-conscious , even making fashion statements at times .

Now-a-days the prince mainly wears navy COLOR suits ITEM (sometimes double-breasted DESIGN) , light blue COLOR button-ups ITEM with classic LOOK pointed DESIGN collars PART , and burgundy COLOR ties ITEM .

Natural Language Processing

One-Hot Encoding

One-Hot Encoding



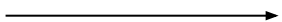
- Esparsidade e dimensionalidade elevada (tamanho do dicionário)
- Nenhuma informação semântica

Bag-of-words
e TF-IDF

Bag-of-words

- Coleção de documentos ou textos e contagem de palavras.
- Sentido de palavras dado pela presença em documentos similares;

It was the best of times,
It was the worst of times,
It was the age of wisdom,
It was the age of foolishness



it
was
the
best
of
times
worst
age
wisdom
foolishness

Bag-of-words

	it	was	the	best	of	times	worst	age	wisdom	foolishness
"it was the best of times"	1	1	1	1	1	1	0	0	0	0
"it was the worst of times"	1	1	1	0	1	1	1	0	0	0
"it was the age of wisdom"	1	1	1	0	1	0	0	1	1	0
"it was the age of foolishness"	1	1	1	0	1	0	0	1	0	1

- Palavras muito comuns podem dominar a construção dos vetores.
- Problema com dimensionalidade (quantidade de documentos) e esparsidade.

TF-IDF

- Term Frequency - Inverse Document Frequency
    - Term Frequency: Quanto a palavra aparece em cada documento
    - Inverse Document Frequency: Quão rara é a palavra nos documentos

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

	it	was	the	best	of	times	worst	age	wisdom	foolishness
"it was the best of times"	0	0	0	0.602	0	0.301	0	0	0	0
"it was the worst of times"	0	0	0	0	0	0.301	0.602	0	0	0
"it was the age of wisdom"	0	0	0	0	0	0	0	0.3	0.602	0
"it was the age of foolishness"	0	0	0	0	0	0	0	0.3	0	0.602

TF-IDF

- Mantém problema da dimensionalidade e da esparsidade
- Não captura muito bem semântica e similaridade de palavras

WORD EMBEDDINGS

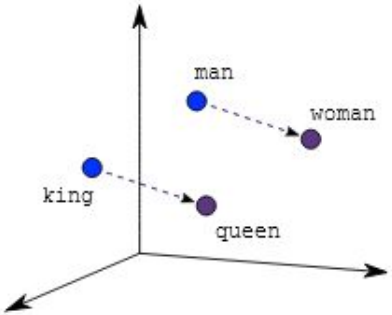
WORD EMBEDDINGS

- Representação n-dimensional (50, 100, 300, 600, 1000) de uma palavra que mantém relações de similaridade.

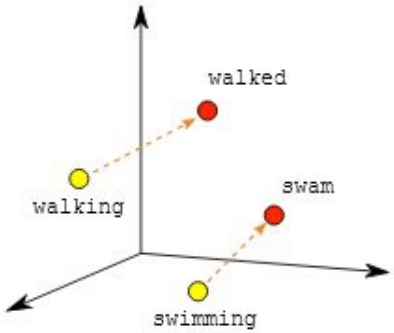
- Embeddings treinados com grandes bases de dados (Wikipedia , Gigaword), sendo que o pressuposto é que o sentido de uma palavra é dependente das palavras que costumam aparecer ao seu redor.

- Maioria dos embeddings (WORD2VEC, FastText, Wang2Vec) é treinado através de uma rede neural que tenta prever uma palavra a partir das palavras ao seu redor (CBOW) ou prever as palavras ao redor a partir da palavra (Skip-gram).

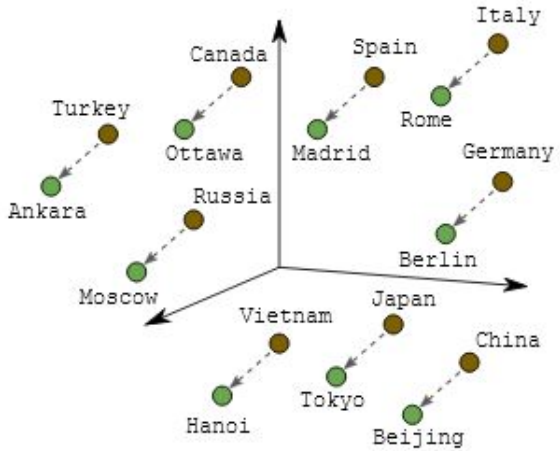
WORD EMBEDDINGS



Male-Female



Verb Tense



Country-Capital

BAIXANDO WORD EMBEDDINGS

Word2Vec

Modelo	Corpora STIL 2017
CBOW 50 dimensões	download
CBOW 100 dimensões	download
CBOW 300 dimensões	download
CBOW 600 dimensões	download
CBOW 1000 dimensões	download
SKIP-GRAM 50 dimensões	download
SKIP-GRAM 100 dimensões	download
SKIP-GRAM 300 dimensões	download
SKIP-GRAM 600 dimensões	download
SKIP-GRAM 1000 dimensões	download

Ver Detalhes »

FastText

Modelo	Corpora STIL 2017
CBOW 50 dimensões	download
CBOW 100 dimensões	download
CBOW 300 dimensões	download
CBOW 600 dimensões	download
CBOW 1000 dimensões	download
SKIP-GRAM 50 dimensões	download
SKIP-GRAM 100 dimensões	download
SKIP-GRAM 300 dimensões	download
SKIP-GRAM 600 dimensões	download
SKIP-GRAM 1000 dimensões	download

Ver Detalhes »

Wang2Vec

Modelo	Corpora STIL 2017
CBOW 50 dimensões	download
CBOW 100 dimensões	download
CBOW 300 dimensões	download
CBOW 600 dimensões	download
CBOW 1000 dimensões	download
SKIP-GRAM 50 dimensões	download
SKIP-GRAM 100 dimensões	download
SKIP-GRAM 300 dimensões	download
SKIP-GRAM 600 dimensões	download
SKIP-GRAM 1000 dimensões	download

Glove

Modelo	Corpora STIL 2017
GLOVE 50 dimensões	download
GLOVE 100 dimensões	download
GLOVE 300 dimensões	download
GLOVE 600 dimensões	download
GLOVE 1000 dimensões	download

Ver Detalhes »

http://nilc.icmc.usp.br/embeddings

Recurrent Neural Networks

Recurrent Neural Networks

★ 10/10

Pure Magic
seremela-1 30 November 2004

This movie is a delight for those of all ages.

I have seen it several times and each time I am enchanted by the characters and magic.

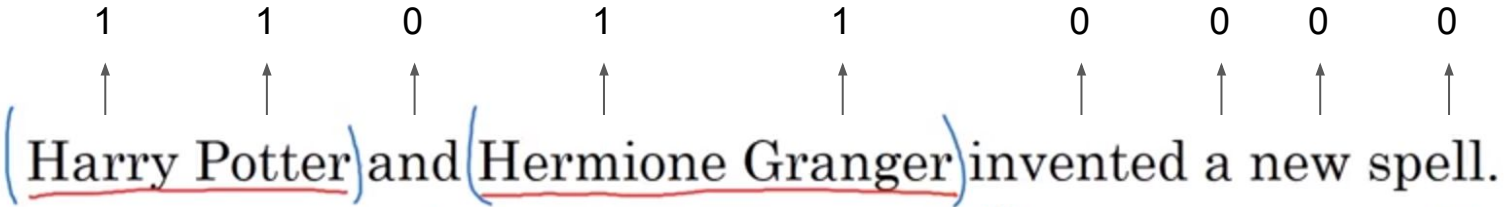
The cast is outstanding, the special effects delightful, everything most believable.

You have young Harry, a mistreated youth who is "Just Harry" to himself. And then, he embarks on a most beautiful adventure to the Hogwarts school.

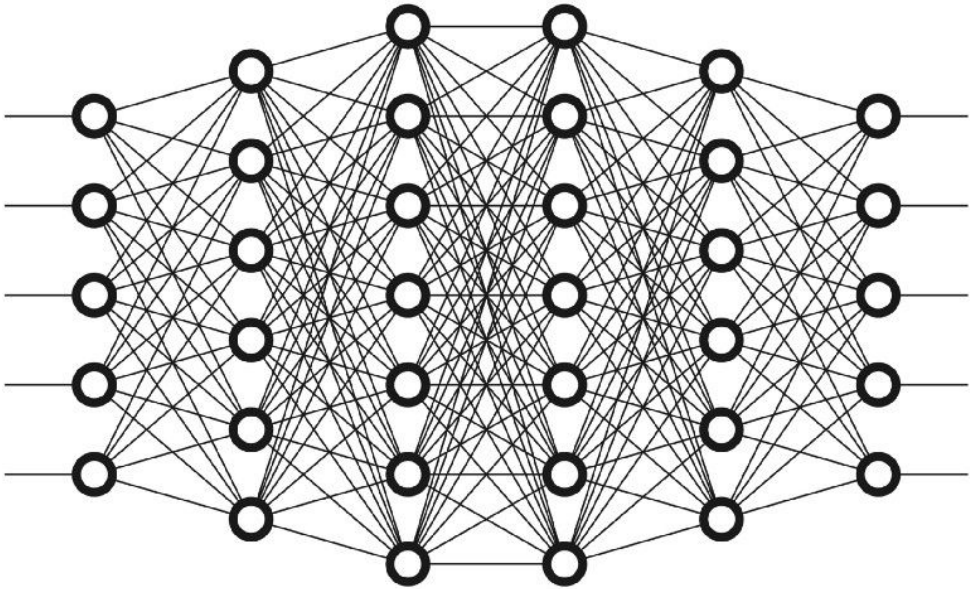
He meets Ron and Hermione, one an adorable mischief maker, the other a very tense and studious young lady.

Together, the trio try to set things right in the school.

It's the ultimate fantasy for young and old.

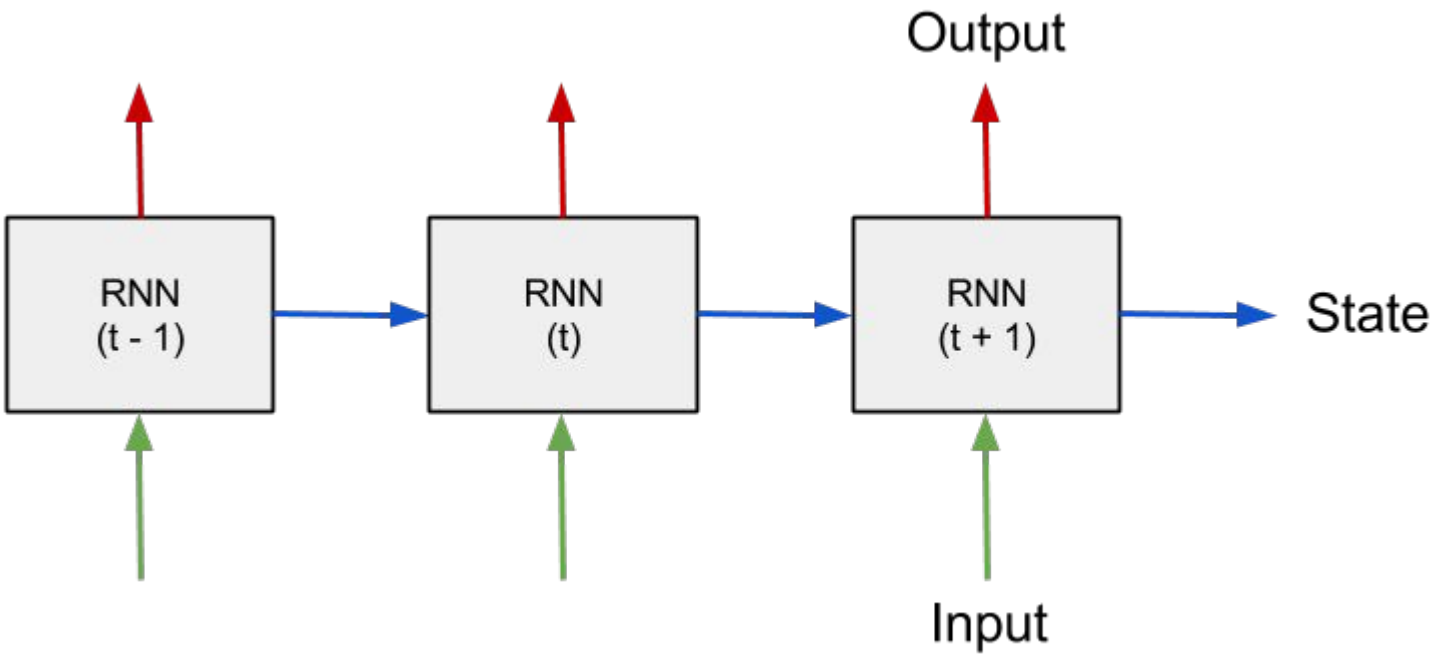


Neural Networks

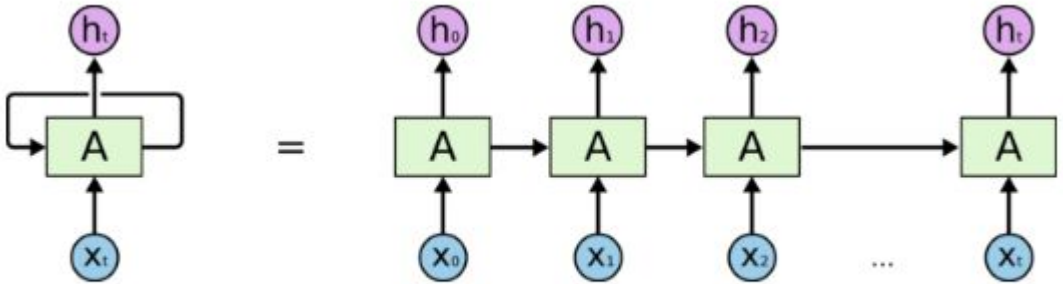


- Muitos pesos / pesos não compartilhados
- Não considera sequência
- Como lidar com inputs de tamanhos diferentes ou outputs n:n?

Recurrent Neural Networks

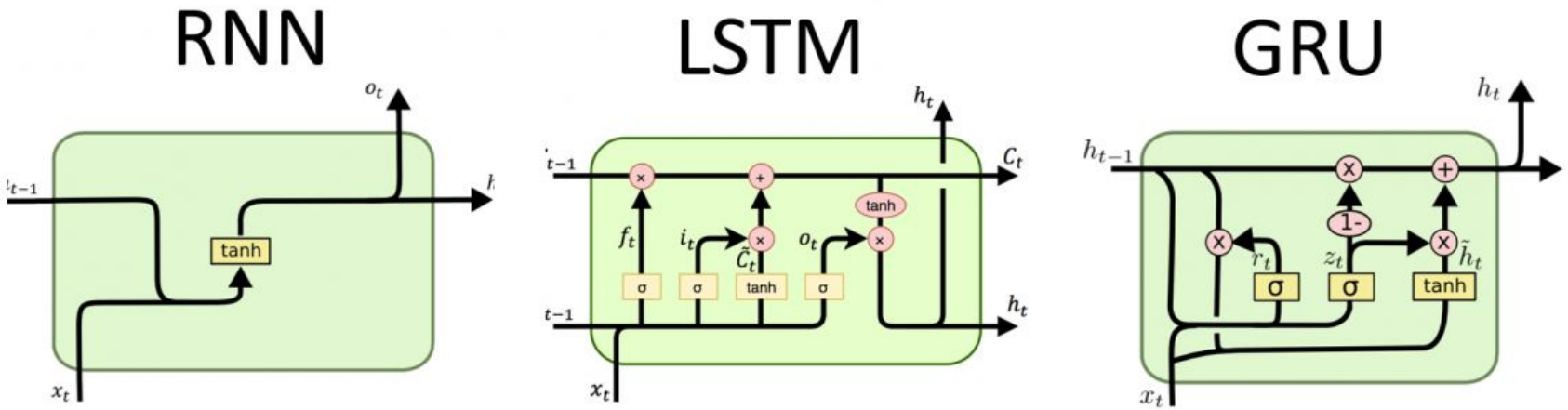


Recurrent Neural Networks



- Pesos compartilhados
- Inputs e Outputs flexíveis
- Problema para sequências longas - Vanishing or exploding gradients

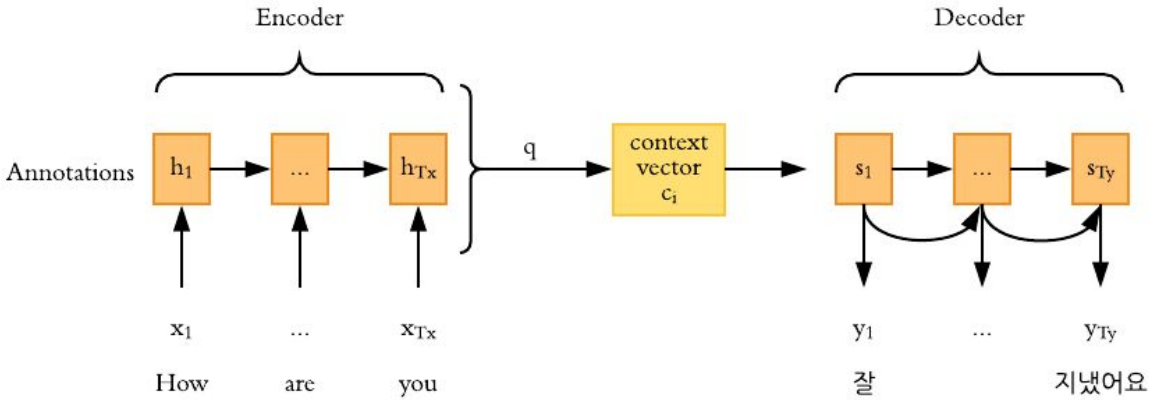
RNN, LSTM E GRU



- Estrutura deveria conseguir manter em memória longas sequências, mas na prática isso não acontece em muitos casos.

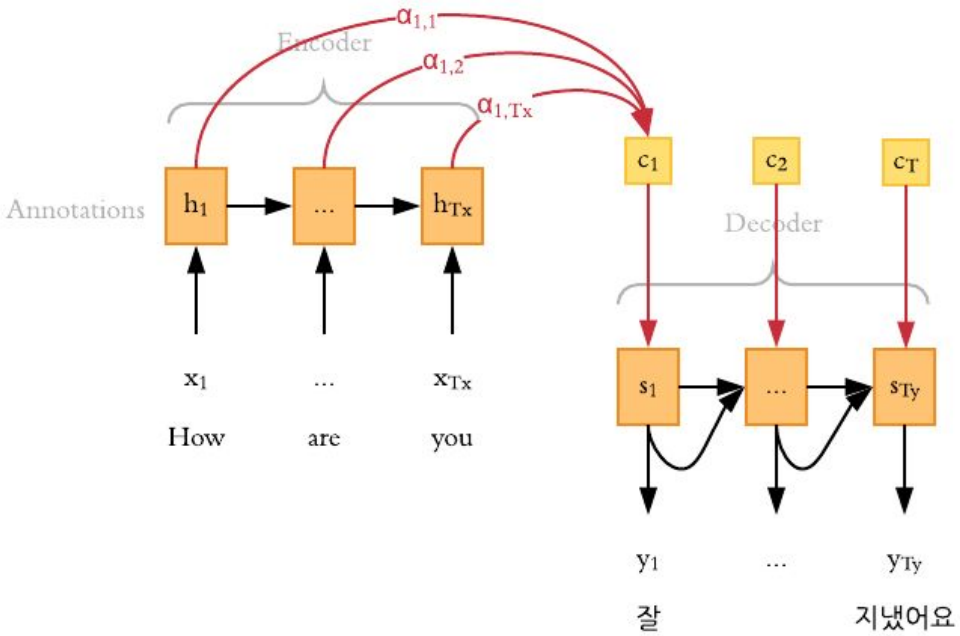
Attention

Attention



- Toda a tradução deve ser feita a partir de um só vetor de contexto (sentence embedding)

Attention



- Com attention, a ideia é gerar um vetor de contexto para cada palavra a ser traduzida, a partir de uma combinação do peso de cada palavra.
- Intuição é como se o modelo aprendesse no que focar sua atenção.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

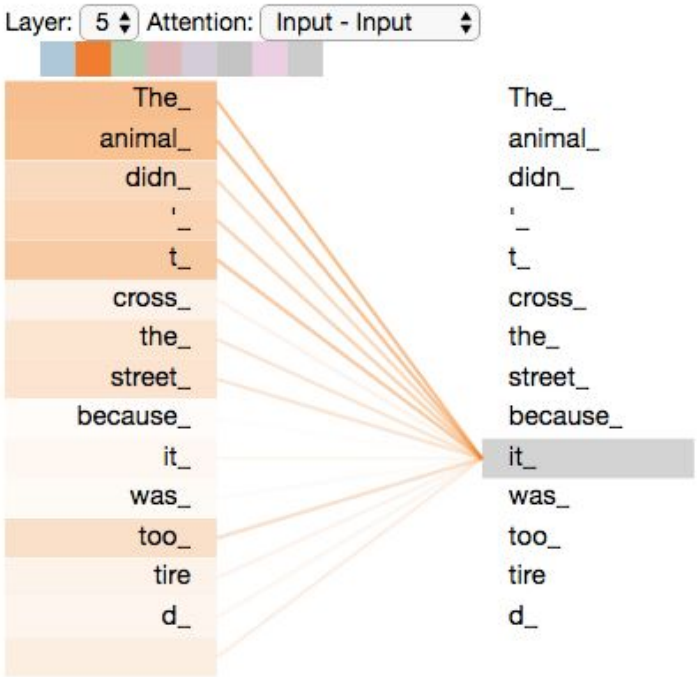
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Self Attention



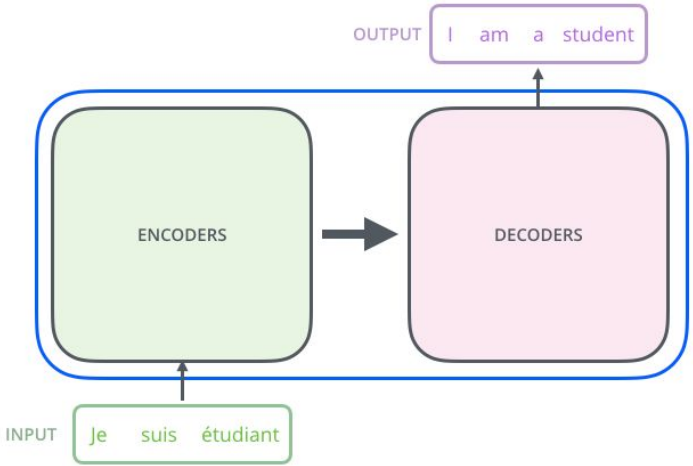
- Ao que a palavra “it” se refere?
- Ao processar uma determinada palavra, faz sentido considerar as demais!

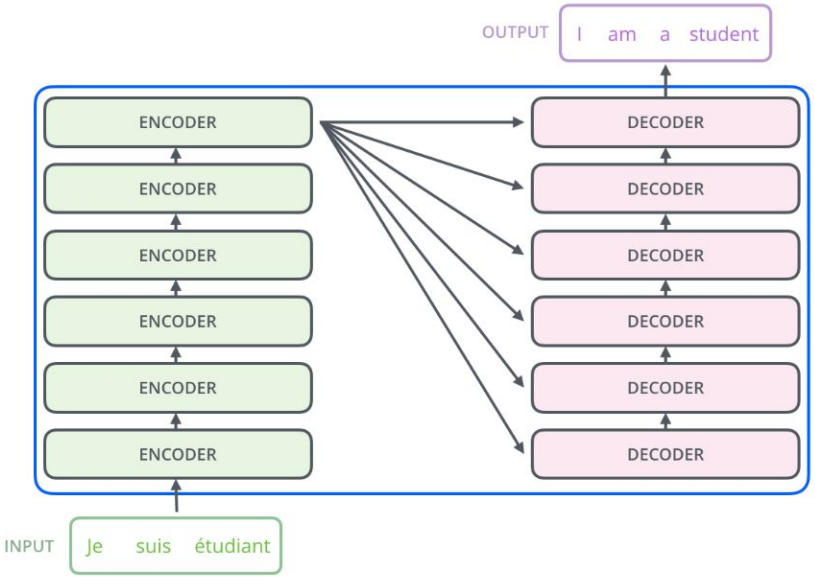
Transformer

Transformer

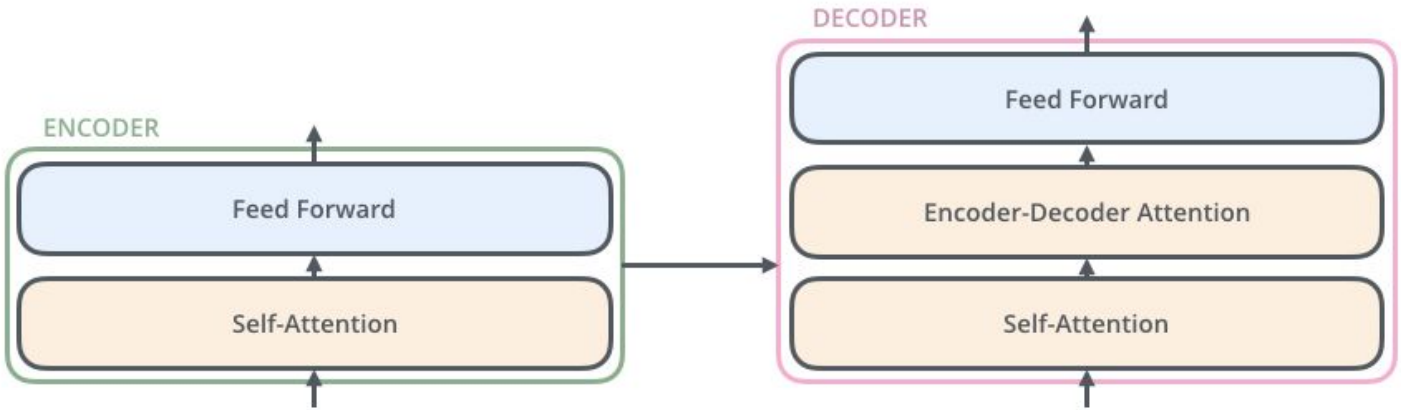
- Em termos simples: um encoder-decoder baseado em mecanismos de (self-)attention

Transformer

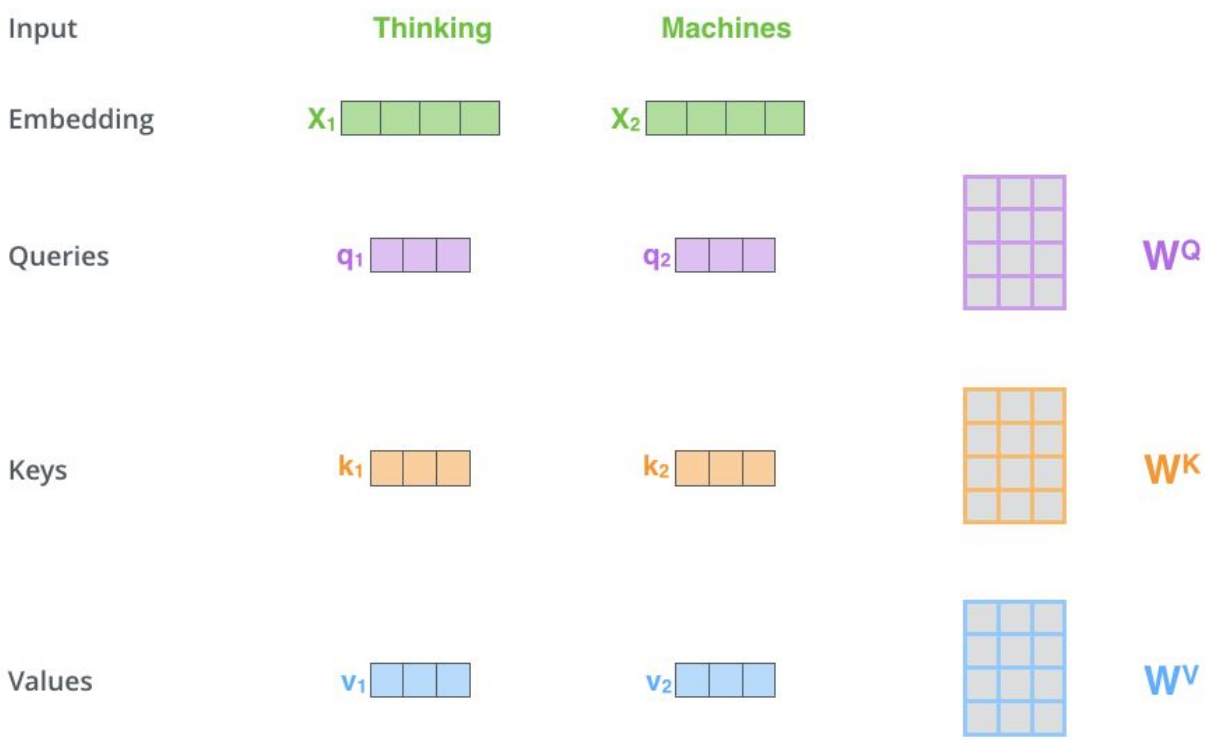




Transformer

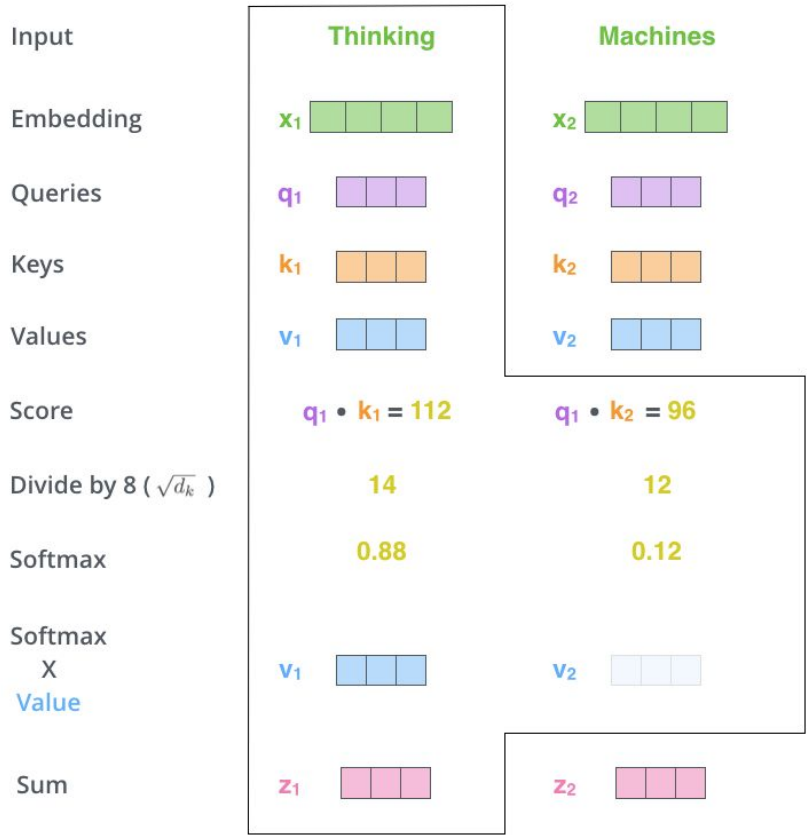


Transformer

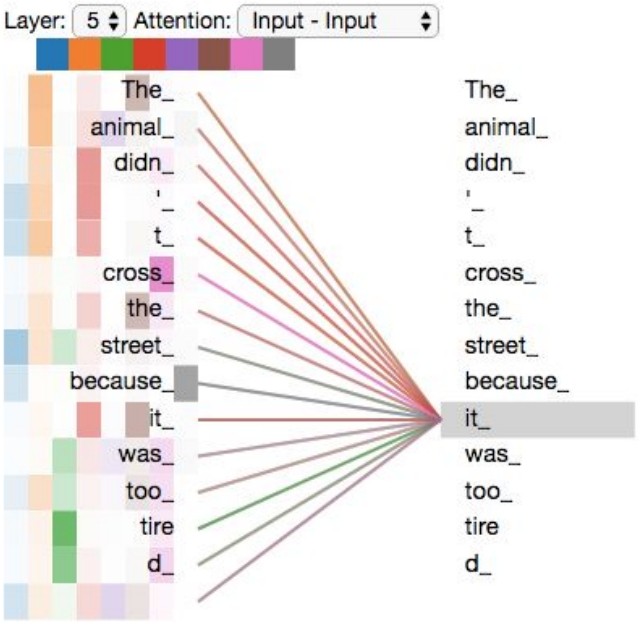


Multiplying x1 by the WQ weight matrix produces q1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence.

Transformer



Transformer
Multi-headed Attention

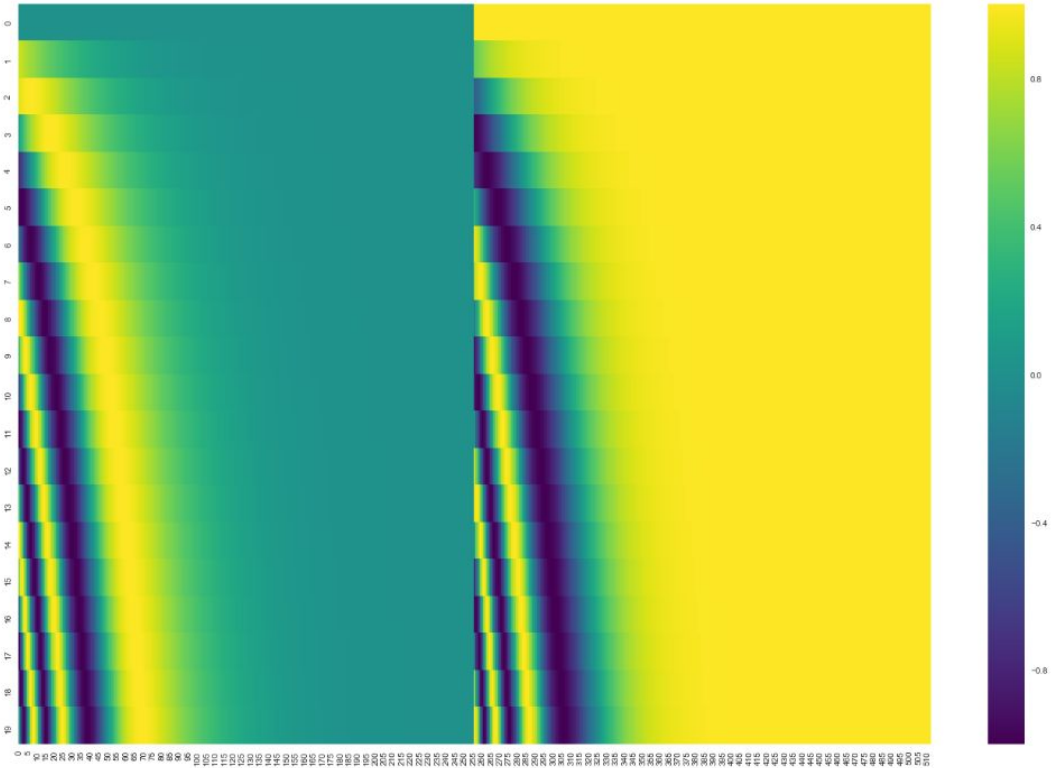


- Mais de uma “cabeça” de atenção
- Permite levar em consideração diferentes aspectos ao relacionar palavras (um por cabeça)

Transformer

- Se não usamos RNN, como a rede sabe qual a ordem das palavras na frase???

Transformer
Positional Encoding



A real example of positional encoding for 20 words (rows) with an embedding size of 512 (columns). You can see that it appears split in half down the center. That's because the values of the left half are generated by one function (which uses sine), and the right half is generated by another function (which uses cosine). They're then concatenated to form each of the positional encoding vectors.

Transformer
Arquitetura Completa

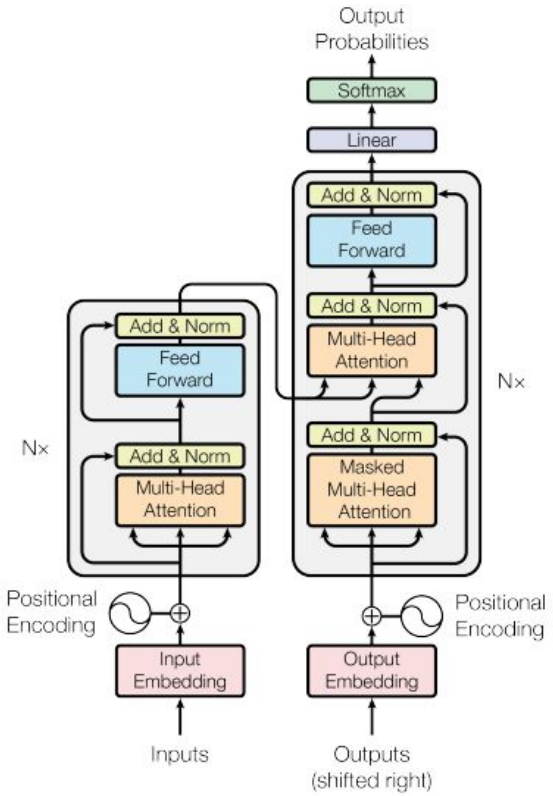


Figure 1: The Transformer - model architecture.

Usos dos Transformers

Análise de Sequências	Encoder
Síntese de Sequências	Decoder
Tradução entre Sequências	Encoder e Decoder

Aplicações

Aplicações
GPT-2 (OpenAI)

Better Language Models and Their Implications

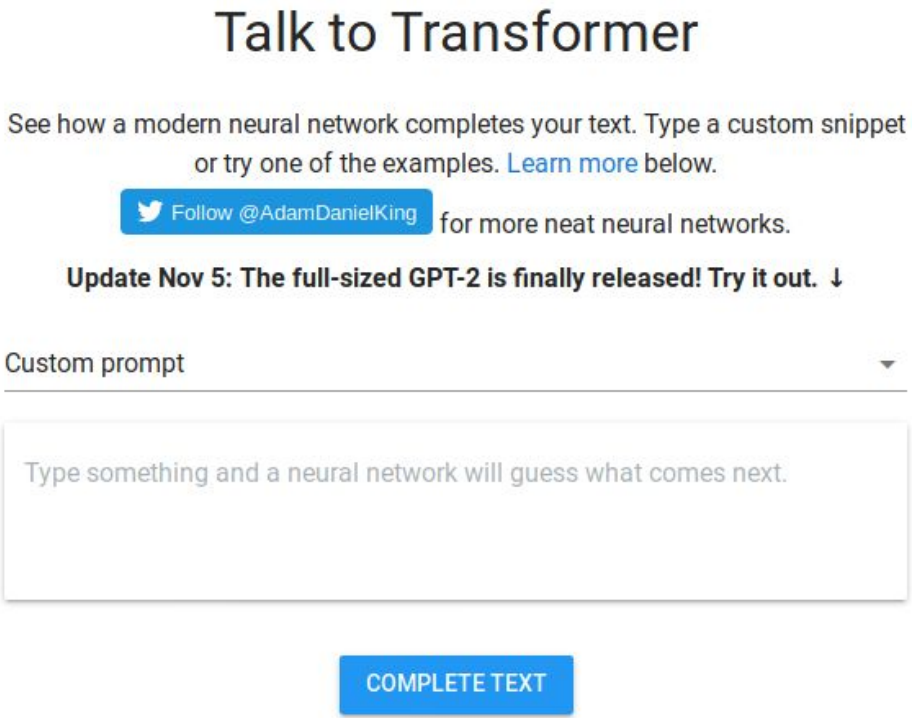
We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization—all without task-specific training.

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

- Treinado para predizer a próxima palavra em 40GB de textos retirados da internet (via Reddit)
- 1.5bi de parâmetros
- Modelo treinado inicialmente mantido em segredo por segurança

Aplicações
Talk to Transformer



Aplicações
BERT (Google)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Understanding searches better than ever before

Pandu Nayak
Google Fellow and Vice President, Search

Published Oct 25, 2019

If there's one thing I've learned over the 15 years working on Google Search, it's that people's curiosity is endless. We see billions of searches every day, and 15 percent of those queries are ones we haven't seen before--so we've built ways to return results for queries we can't anticipate.

- *Propósito: Transfer learning*

Aplicações
Resolvendo Equações

ANALYSING MATHEMATICAL REASONING ABILITIES OF NEURAL MODELS

David Saxton
DeepMind
saxton@google.com

Edward Grefenstette
DeepMind
egrefen@fb.com

Felix Hill
DeepMind
felixhill@google.com

Pushmeet Kohli
DeepMind
pushmeet@google.com

Easiest maths for neural networks The easiest question types were finding the place value in a number, and rounding decimals and integers, which all models got nearly perfect scores on. Questions involving comparisons also tended to be quite easy, possible because such tasks are quite perceptual (e.g. comparing lengths or individual digits). This success includes questions with module composition, for example `Let k(c) = -611*c + 2188857. Is k(-103) != 2251790?` (False) and mixtures of decimals and rationals, for example, `Sort -139/4, 40.8, -555, 607 in increasing order`. Overall it seems that magnitude is easy for neural networks to learn.

Hardest maths for neural networks Perhaps not surprisingly, some of the hardest modules include more number-theoretic questions which are also hard for humans, such as detecting primality and factorization. The Transformer model still gives plausible-looking answers, such as factoring `235232673` as `3, 11, 13, 19, 23, 1487` (the correct answer is `3, 13, 19, 317453`).

	Parameters	Interpolation	Extrapolation
Simple **LSTM**	18M	0.57	0.41
Simple **RMC**	38M	0.53	0.38
Attentional **LSTM**, LSTM encoder	24M	0.57	0.38
Attentional **LSTM**, bidir LSTM encoder	26M	0.58	0.42
Attentional **RMC**, bidir LSTM encoder	39M	0.54	0.43
Transformer	30M	**0.76**	**0.50**

Figure 3: Model accuracy (probability of correct answer) averaged across modules. **RMC** is the relational recurrent neural network model.

Aplicações
Resolvendo Equações

DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Anonymous authors
Paper under double-blind review

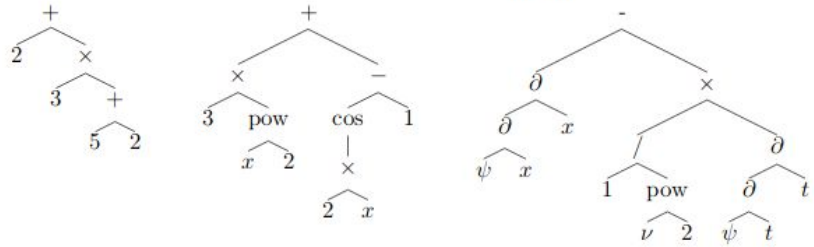
ABSTRACT

Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing these mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica.

On all tasks, we observe that our model significantly outperforms Mathematica. On function integration, our model obtains close to 100% accuracy, while Mathematica does not reach 80%. On first order differential equations, Mathematica is on par with our model when it uses a beam size of 1, i.e. with greedy decoding. However, using a beam search of size 50 our model accuracy goes from 81.2% to 97.0%, largely surpassing Mathematica. Similar observations can be made for second order differential equations, where beam search is even more critical since the number of equivalent solutions is larger. For Mathematica, the performance naturally increases with the timeout duration. For each task, using a timeout of 30 seconds improves the test accuracy by about 4% over a timeout of 10 seconds. Our model, on the other hand, typically finds a solution in less than a second, even with a large beam size. On average, Matlab has a slightly lower performance than Mathematica on the problems we tested.

2.1 EXPRESSIONS AS TREES

Mathematical expressions can be represented as trees, with operators and functions as internal nodes, operands as children, and numbers, constants and variables as leaves. The following trees represent expressions $2 + 3 \times (5 + 2)$, $3x^2 + \cos(2x) - 1$, and $\frac{\partial^2 \psi}{\partial x^2} - \frac{1}{\nu^2} \frac{\partial^2 \psi}{\partial t^2}$:



Equation	Solution
$y' = \frac{16x^3 - 42x^2 + 2x}{(-16x^8 + 112x^7 - 204x^6 + 28x^5 - x^4 + 1)^{1/2}}$	$y = \sin^{-1}(4x^4 - 14x^3 + x^2)$
$3xy\cos(x) - \sqrt{9x^2\sin(x)^2 + 1}y' + 3y\sin(x) = 0$	$y = c\exp\left(\sinh^{-1}(3x\sin(x))\right)$
$4x^4yy'' - 8x^4y'^2 - 8x^3yy' - 3x^3y'' - 8x^2y^2 - 6x^2y' - 3x^2y'' - 9xy' - 3y = 0$	$y = \frac{c_1 + 3x + 3\log(x)}{x(c_2 + 4x)}$

Table 3: Examples of problems that our model is able to solve, on which Mathematica and Matlab were not able to find a solution. For each equation, our model finds a valid solution with greedy decoding.

Aplicações

Música

MuseNet

We've created MuseNet, a deep neural network that can generate 4-minute musical compositions with 10 different instruments, and can combine styles from country to Mozart to the Beatles. MuseNet was not explicitly programmed with our understanding of music, but instead discovered patterns of harmony, rhythm, and style by learning to predict the next token in hundreds of thousands of MIDI files. MuseNet uses the same general-purpose unsupervised technology as GPT-2, a large-scale transformer model trained to predict the next token in a sequence, whether audio or text.

- https://colab.research.google.com/notebooks/magenta/piano_transformer/piano_transformer.ipynb
- https://soundcloud.com/marceloprates/tensor-da-cor-do-mar

Modelos Pré-treinados

State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

🤗 Transformers (formerly known as `pytorch-transformers` and `pytorch-pretrained-bert`) provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch.

- Transformers estado-da-arte pré-treinados
- github.com/huggingface/transformers

🤗 Transformers currently provides 10 NLU/NLG architectures:

1. **BERT** (from Google) released with the paper BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.
2. **GPT** (from OpenAI) released with the paper Improving Language Understanding by Generative Pre-Training by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
3. **GPT-2** (from OpenAI) released with the paper Language Models are Unsupervised Multitask Learners by Alec Radford*, Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskever**.
4. **Transformer-XL** (from Google/CMU) released with the paper Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov.
5. **XLNet** (from Google/CMU) released with the paper XLNet: Generalized Autoregressive Pretraining for Language Understanding by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le.
6. **XLM** (from Facebook) released together with the paper Cross-lingual Language Model Pretraining by Guillaume Lample and Alexis Conneau.
7. **RoBERTa** (from Facebook), released together with the paper a Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov.
8. **DistilBERT** (from HuggingFace), released together with the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter by Victor Sanh, Lysandre Debut and Thomas Wolf. The same method has been applied to compress GPT2 into DistilGPT2.
9. **CTRL** (from Salesforce) released with the paper CTRL: A Conditional Transformer Language Model for Controllable Generation by Nitish Shirish Keskar*, Bryan McCann*, Lav R. Varshney, Caiming Xiong and Richard Socher.
10. Want to contribute a new model? We have added a **detailed guide and templates** to guide you in the process of adding a new model. You can find them in the `templates` folder of the repository. Be sure to check the contributing guidelines and contact the maintainers or open an issue to collect feedbacks before starting your PR.

Perguntas?

Obrigado!

- **Luiz Nonenmacher**

  ljuniornone@gmail.com
  linkedin.com/in/luiz-nonenmacher

- **Marcelo Prates**

  marceloorp@gmail.com
  linkedin.com/in/marceloprates