FIAP

MBA+ | Artificial Intelligence & Machine Learning


AI

Attention

Attention

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

Xiv:1706.03762v5 [cs.CL] 6 Dec 2017

Abstract

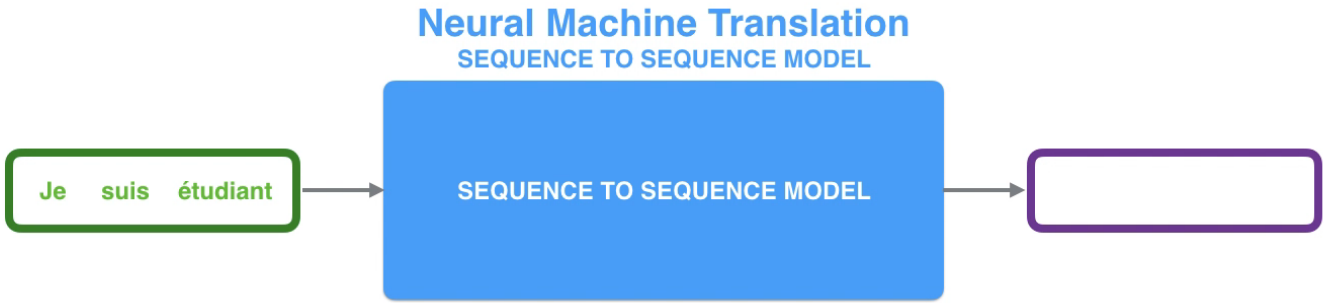
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, 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.

Attention

As Redes Neurais Recorrentes (RNR) obtiveram bastante sucesso ultimamente para tratar dados sequenciais, como texto (tradução), áudio (identificação de fala) e séries temporais (predição). Modelos sequência-para-sequência (abreviado como seq2seq, do inglês sequence to sequence) são uma classe especial de arquitetura das RNR. Eles são usados para resolver problemas complexos como tradução, chat-bots e resumo de textos.

Attention

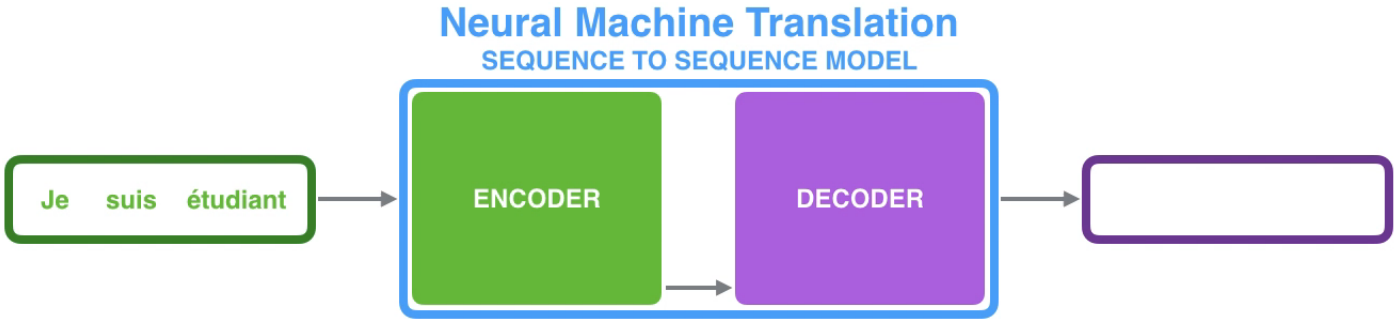
Attention


Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL
Je suis étudiant
SEQUENCE TO SEQUENCE MODEL

Attention

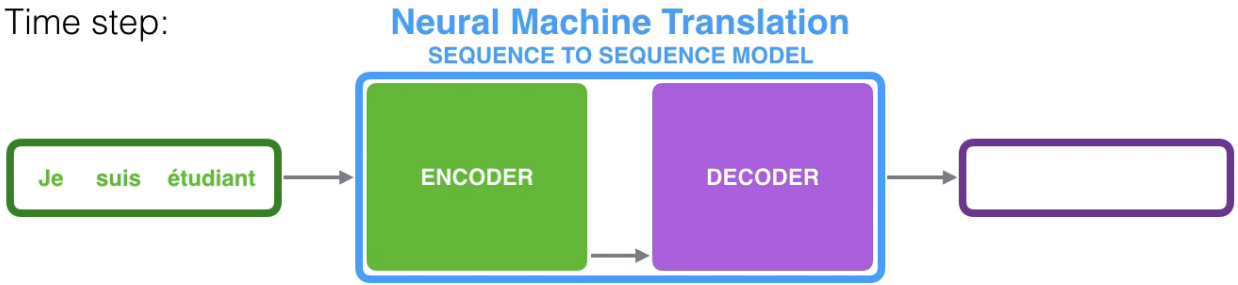
Os modelos seq2seq normalmente são formados por uma arquitetura encoder-decoder, onde o encoder processa a sequência de entrada e comprime essa informação em um vetor de contexto de tamanho fixo (o último hidden state).

Attention


Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL
Je suis étudiant
ENCODER
DECODER

Attention

Time step:

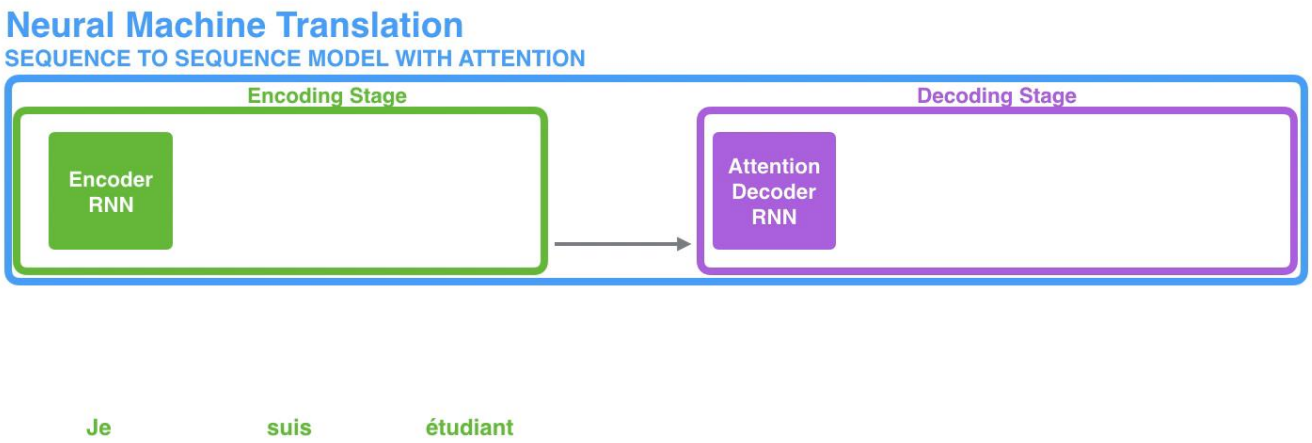


Attention

1) As RNR usam um tamanho fixo para a representação da entrada e saída. Porém, em muitos casos, o tamanho da entrada e saída é diferente. Tome como exemplo a tradução inglês-português: “I did my homework yesterday” (5 palavras) vira “Eu fiz o meu dever de casa ontem” (7 palavras).

2) O uso de um vetor de sequência de tamanho fixo torna-se problemático para sequências de entrada longas, visto que o vetor de contexto deve considerar a informação de toda a sequência de entrada. Na tradução de uma frase longa, por exemplo, há grandes chances da primeira palavra de uma frase em inglês ser altamente correlacionada com a primeira palavra da respectiva tradução em português. Logo, o vetor de contexto teria de ser capaz de representar essa longa dependência no tempo.

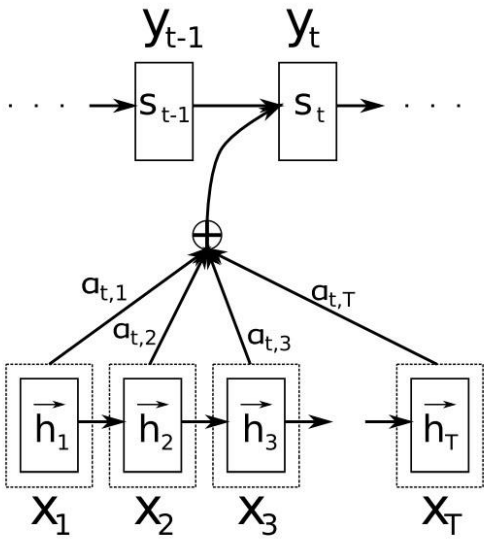
Attention


Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL WITH ATTENTION
Encoding Stage
Decoding Stage
Encoder RNN
Attention Decoder RNN
Je
suis
étudiant

Attention

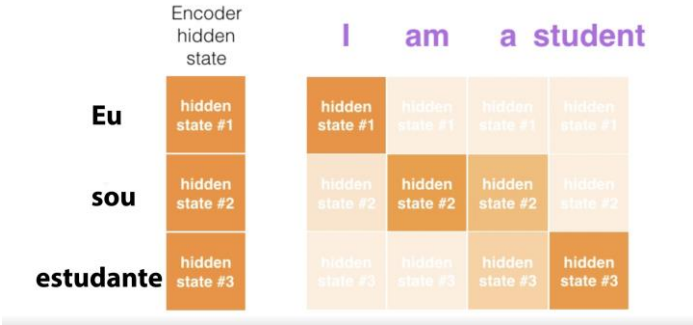
O mecanismo de atenção surge para suprir essa limitação do vetor de contexto. Em vez de codificar toda a sequência de entrada em um vetor de contexto, o mecanismo de atenção permite o decoder “focar” em diferentes partes da sequência de entrada em cada etapa da geração da saída. Ou seja, o decoder utiliza todos os estados intermediários para gerar a saída.

Attention



A imagem representa o mecanismo de atenção, x é a sequência de entrada e y a de saída. É importante notar que agora a saída depende da combinação ponderada de todos os estados de entrada. Os a's representam os pesos de atenção, ou seja, quanto cada estado de entrada é considerado em cada saída. Por exemplo, se at,1 é um valor grande, isso significa que a primeira palavra de entrada é importante na geração da segunda palavra na saída.

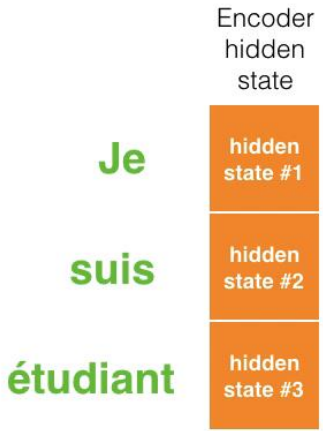
Attention



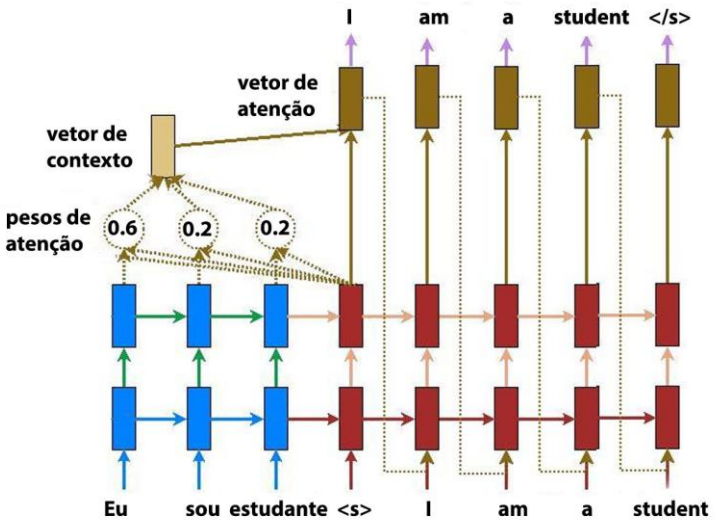
Uma das vantagens de utilizar o mecanismo de atenção é a interpretabilidade. É possível saber qual parte da entrada foi mais relevante para cada parte da saída.

A imagem acima mostra a tradução da frase “eu sou estudante” para o inglês. Valores maiores de atenção são representados por cores mais escuras. Perceba que “eu” possui um peso grande para “I” assim como “estudante” possui para “student”.

Attention


Encoder
hidden
state
Je
suis
étudiant
hidden
state #1
hidden
state #2
hidden
state #3

Attention



A arquitetura é similar ao encoder-decoder básico, com uma particularidade entre o encoder (azul) e o decoder (vermelho) representados na imagem. A diferença principal em relação aos outros modelos seq2seq é o vetor de contexto que considera todo os elementos da entrada. Cada elemento da saída considera o respectivo vetor de contexto e a saída no instante de tempo anterior

Attention

A atenção é definida pelas três equações:

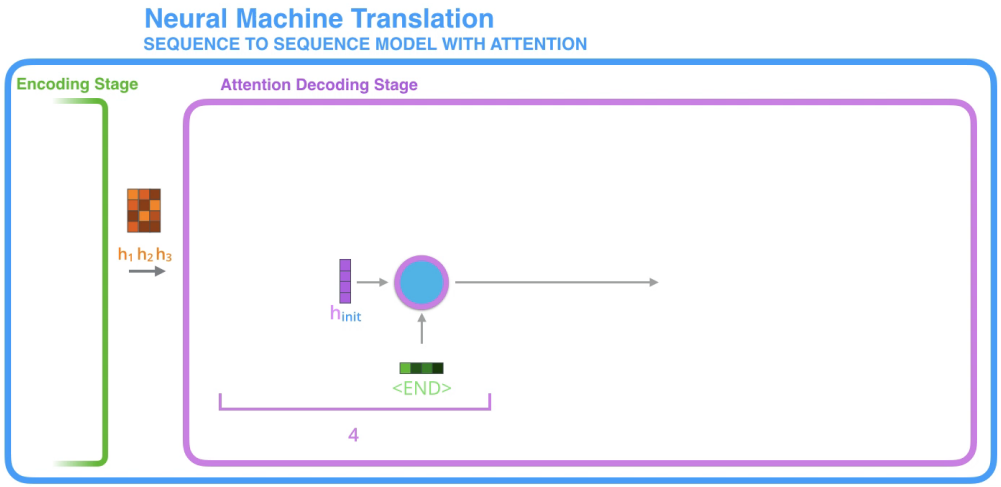
$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Pesos de atenção]} \quad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Vetor de contexto]} \quad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Vetor de atenção]} \quad (3)$$

O *score* na equação (1) é normalmente uma rede neural simples, os pesos de atenção são normalizados usando *softmax* em relação ao *input*. O vetor de contexto é calculado pela soma ponderada dos *hidden states* em relação aos seus pesos. Esse vetor é calculado para cada palavra na saída. Se considerarmos todos os pesos de atenção, teremos uma matriz de dimensões N x M, onde N é o tamanho do *input* e M o tamanho do *output*. Por último, o modelo utiliza o vetor de contexto e o *hidden state* para determinar a saída. A equação (3) mostra a tanh como a não linearidade mas pode-se usar outras funções, como ReLU.

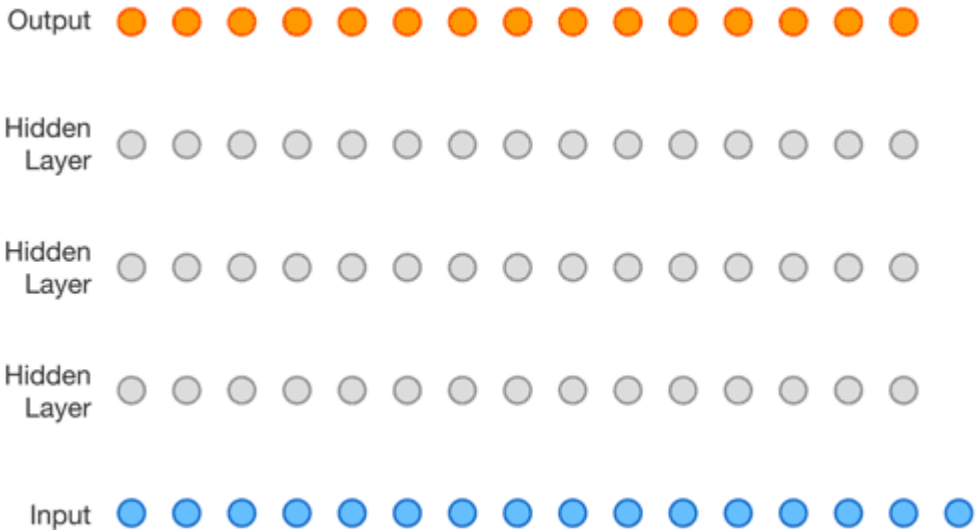
Attention



Vamos agora juntar tudo na visualização ao lado e ver como o processo de atenção funciona:

O decodificador de atenção RNN absorve a incorporação do token <END> e um estado oculto do decodificador inicial.
A RNN processa suas entradas, produzindo uma saída e um novo vetor de estado oculto (h4). A saída é descartada.
Etapa de atenção: Utilizamos os estados ocultos do codificador e o vetor h4 para calcular um vetor de contexto (C4) para este passo no tempo.
Concatenamos h4 e C4 em um vetor.
Passamos esse vetor através de uma rede neural feedforward (uma treinada em conjunto com o modelo).
A saída da rede neural feedforward indica a palavra de saída desta etapa do tempo.
Repita para as próximas etapas

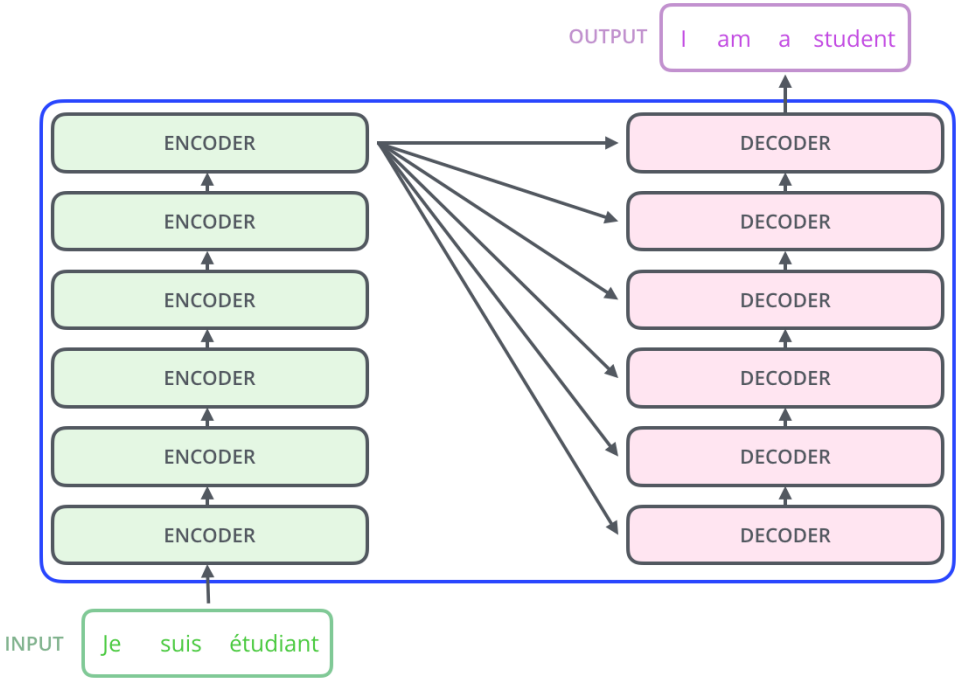
Attention



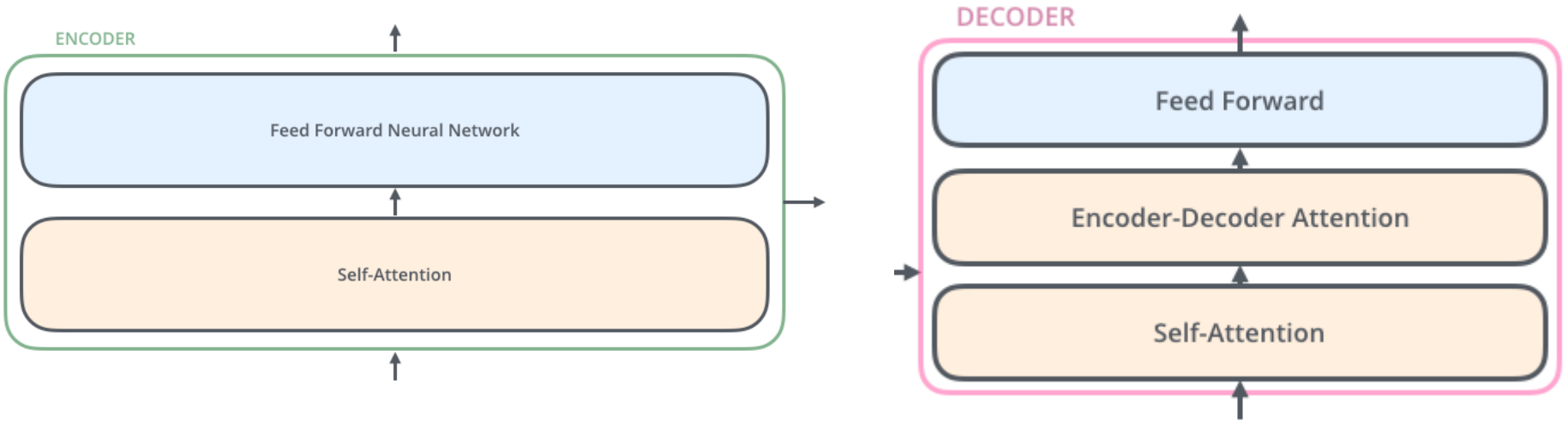
Transformer

Transformer

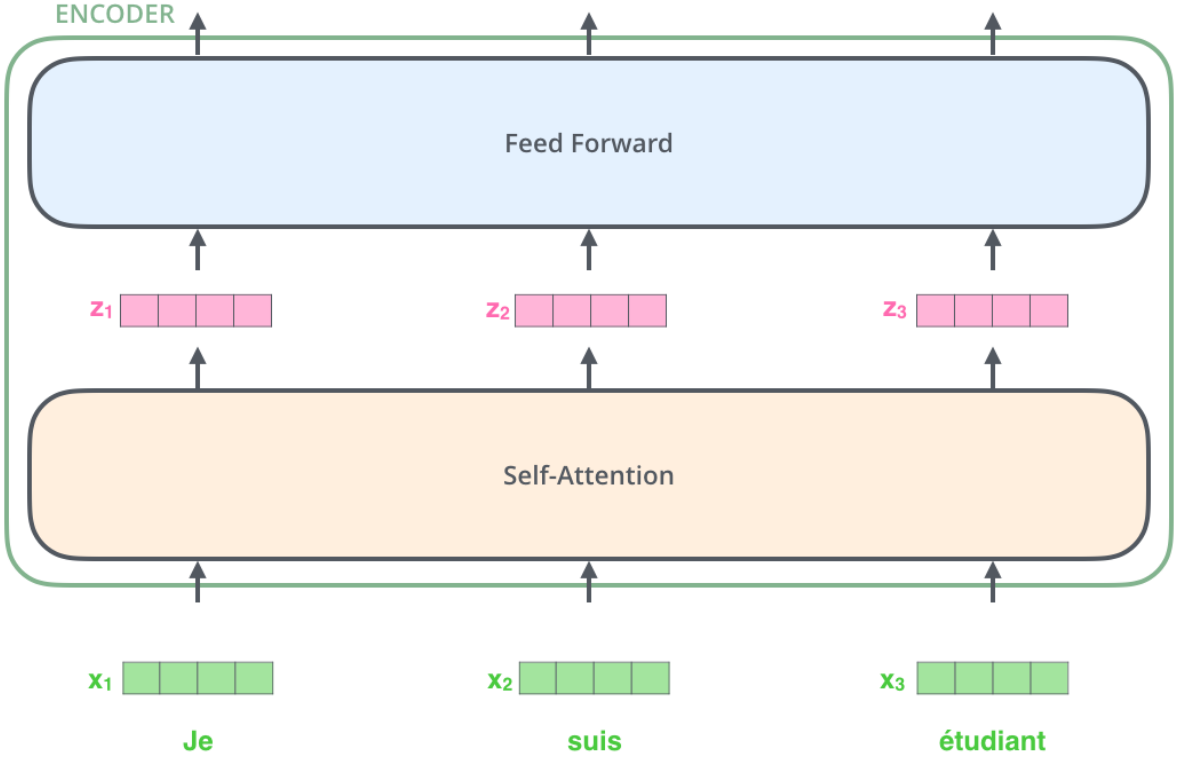
Transformer



Transformer

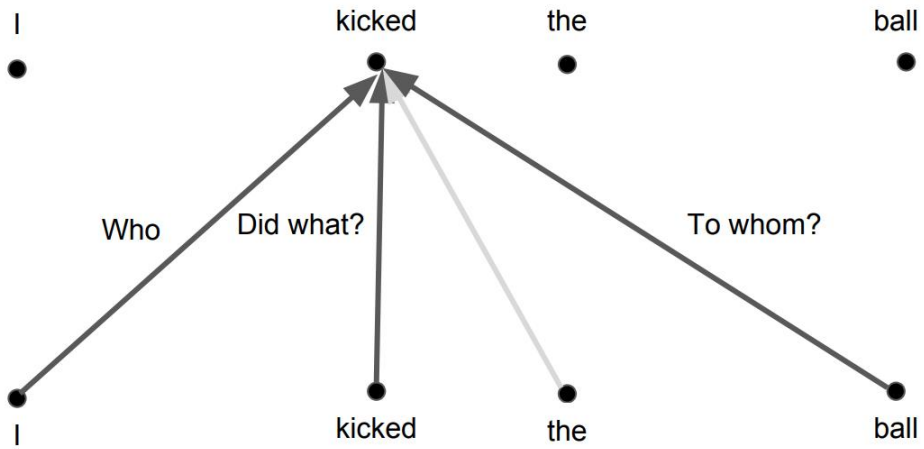


Transformer



Transformer

Self-Attention



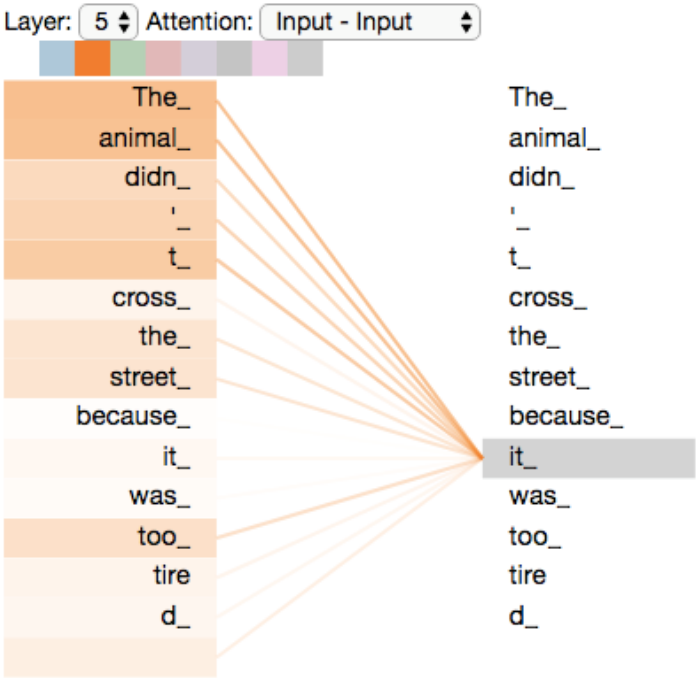
Transformer

“Eu vi um homem na montanha com um telescópio”

1. ***Eu vi um homem. O homem estava na montanha. Eu estava com o telescópio.***
2. ***Eu vi um homem. O homem estava na montanha. Eu estava com o telescópio.***
3. ***Eu vi um homem. O homem estava na montanha. O homem estava com o telescópio.***
4. ***Eu vi um homem. Eu estava na montanha. Eu estava com o telescópio.***

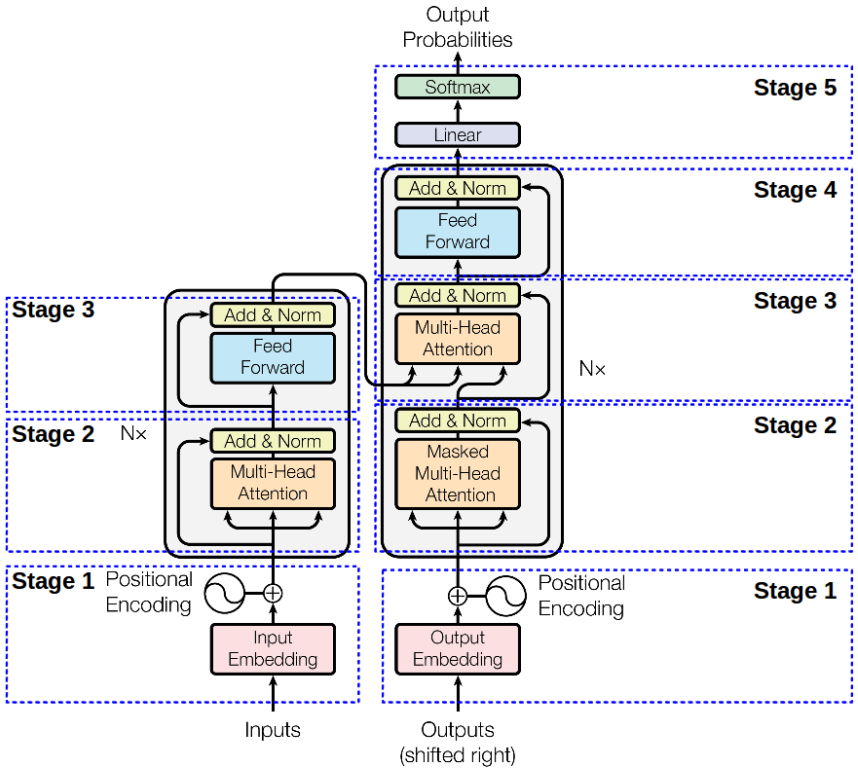
Transformer

The animal didn't cross the street because it was too tired



I grew up in France… I speak fluent …

Transformer



Questions and Feedback

Thank you!

Obrigado !

Vinicius Fernandes Caridá
vfcarida@gmail.com

@Vinicius Caridá

@Vinicius Caridá
@machine learning Brasil

@vfcarida

Copyright © **2018 Prof. Vinicius Fernandes Caridá**
Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proíbido sem o consentimento formal, por escrito, do Professor (autor).