MICAS913 : Deep Learning

Transformers, prepared for Prof. Yousefi MANSOOR

Ali El Hadi ISMAIL FAWAZ Valentin GORSE

Institut Polytechnique de Paris, Master year 2
Machine Learning, Communication and Security

Graduate year - February 16, 2022

Table of Contents

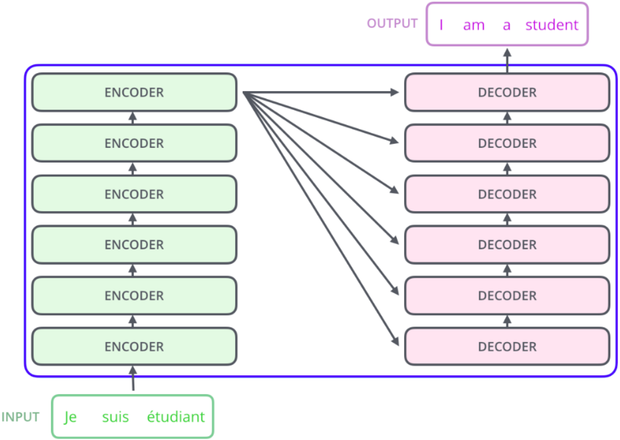
Table of Contents

Introduction

Attention is All you Need (Google Brain 2017)

- Machine translation
- Attention Mechanism
- Encoder , Decoder
- Embedding
- Residual Connections
- Stack encoding decoding

Introduction - Global Concept



A very important note :

The number of encoder and decoders should be the same.

Outline

Table of Contents

Encoder

Each encoder consists of :

- Self attention layer
- Feed Forward Network

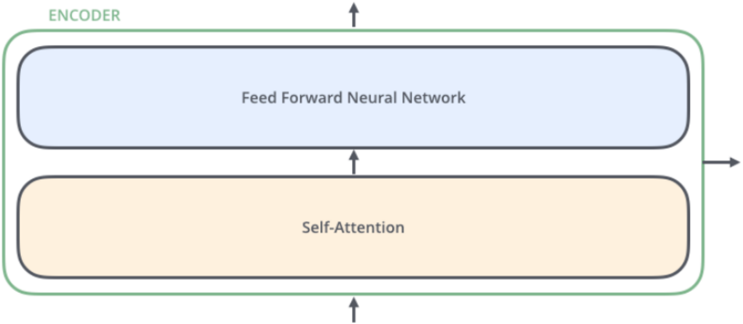


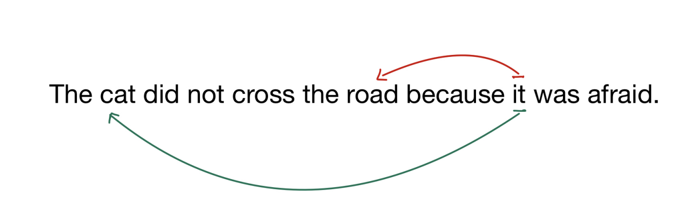
Table of Contents

Self Attention Mechanism

Motivation

Its job is to look at other words in the input sentence while it encodes a specific word of that sentence.

Example



- Human's answer : "it" refers to "cat"
- Computer's answer : "it" can refer to "road"

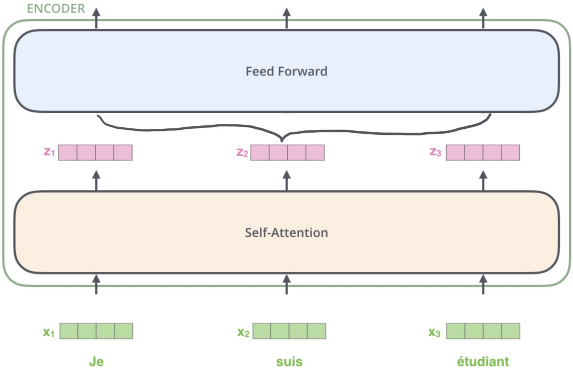
Self Attention Mechanism - Analogy with Human brain

Self Attention Mechanism - Embedding

Similar to any NLP procedure , we embed our input words.

Each word is embedded into a vector of size 512. We'll represent those vectors with these simple boxes.

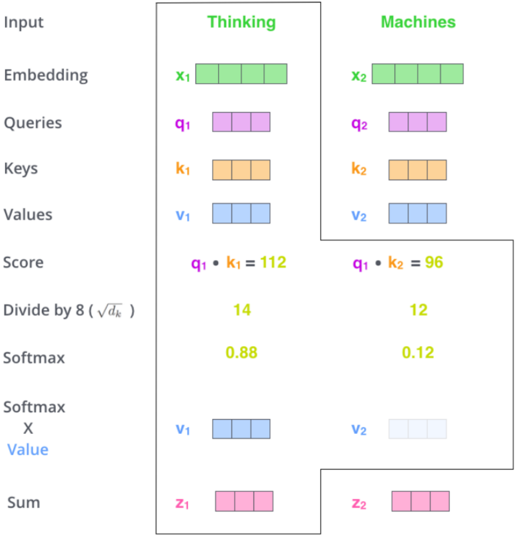
Self Attention Mechanism - Transformers Properties



2 properties :

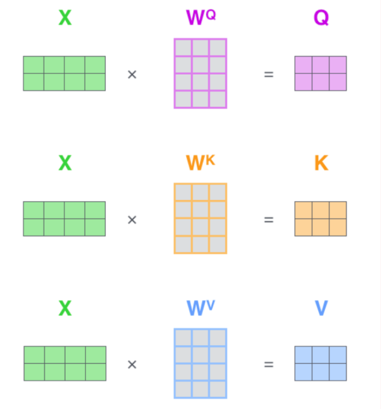
- Path dependencies in self attention
- Path independencies in feed forward network

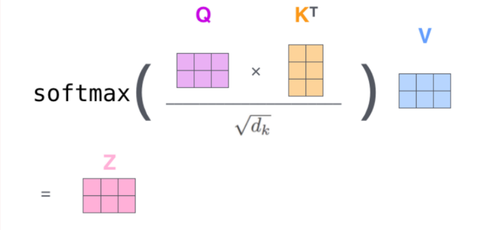
Self Attention Mechanism - In detail



Self Attention Mechanism - In detail (2)

Of course we use MATRICES





The self-attention calculation in matrix form

Remark

The complexity of the self-attention mechanism is $O(n^2.d)$

Self Attention Mechanism - Multi Head Attention

Motivation

- It expends the model's ability to focus on different positions.
- It gives the attention layer multiple representation subspaces.

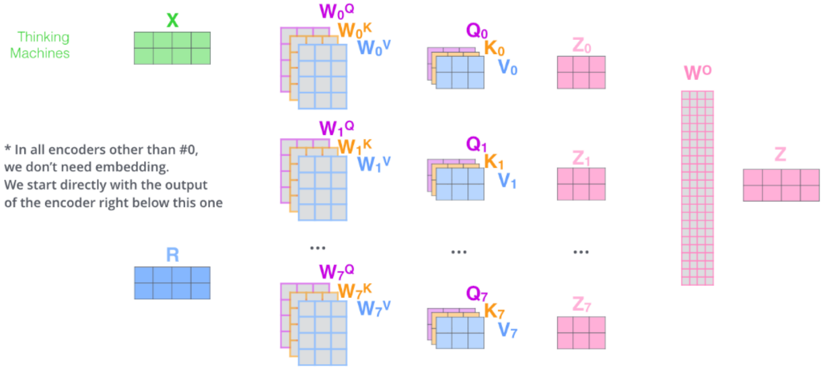
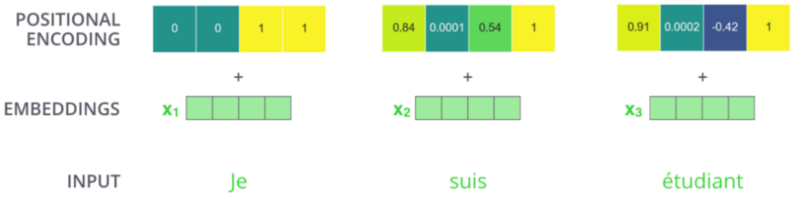


Table of Contents

Positional Encoding



A real example of positional encoding with a toy embedding size of 4

Function of PE used in Attention is all you need :

$$PE(pos, i) = \begin{cases} \sin(\omega_k.pos) & \text{if } i = 2k \\ \cos(\omega_k.pos) & \text{if } i = 2k+1 \\ s.t. \quad w_k = 10000^{-2k/d} \end{cases}$$

d is the size of the embeddings, *pos* is the position of the word in the sentence and i is the position of the embedding of the word *pos*.

Table of Contents

Residual Connections - Recap Encoder

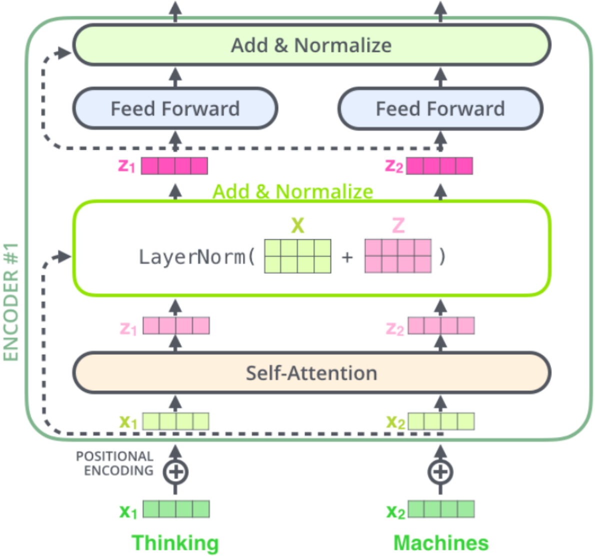
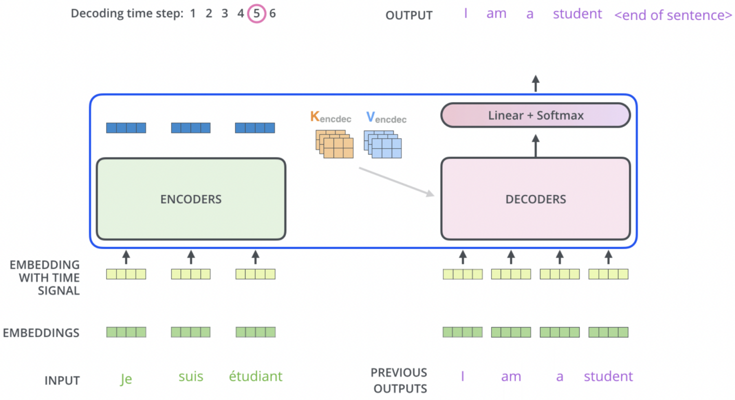


Table of Contents

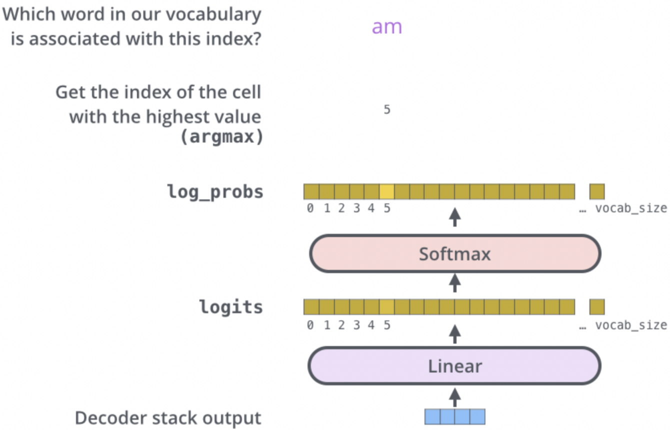
Decoders



Masking in self-attention calculation

The decoded embedding at position i can only see the decoded words from position 0 to i-1. We attribute $-\infty$ value to none visible positions before the Softmax.

Decoders - The Final Layer



This figure starts from the bottom with the vector produced as the output of the decoder stack. It is then turned into an output word.

Table of Contents

Conclusion

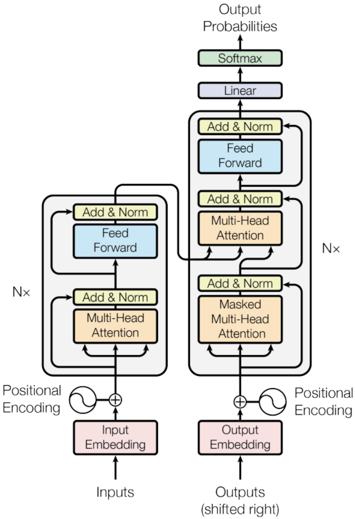


Figure 1: The Transformer - model architecture.

Table of Contents

Implementation

Now we go to our notebook ...

Appendix - Proof of positional encoding

We want to proove that For every sine-cosine pair corresponding to frequency ω_i, there is a linear transformation $M \in \mathbb{R}^{2x2}$ (indep of t) where the following equation holds:

$$M.\begin{pmatrix} sin(\omega_k.pos) \\ cos(\omega_k.pos) \end{pmatrix} = \begin{pmatrix} sin(\omega_k(pos+\alpha)) \\ cos(\omega_k(pos+\alpha)) \end{pmatrix}$$

Proof :

We can extend the sinus and cosinus :

$$\begin{pmatrix} sin(\omega_k(pos+\alpha)) \\ cos(\omega_k(pos+\alpha)) \end{pmatrix} = \begin{pmatrix} sin(\omega_k.pos)cos(\omega_k.\alpha) + cos(\omega_i.pos)sin(\omega_k.\alpha) \\ cos(\omega_k.pos)cos(\omega_k.\alpha) - sin(\omega_k.pos)sin(\omega_k.\alpha) \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}.\begin{pmatrix} sin(\omega_k.pos) \\ cos(\omega_k.pos) \end{pmatrix} = \begin{pmatrix} sin(\omega_k.pos)cos(\omega_k.\alpha) + cos(\omega_k.pos)sin(\omega_k.\alpha) \\ cos(\omega_k.pos)cos(\omega_k.\alpha) - sin(\omega_k.pos)sin(\omega_k.\alpha) \end{pmatrix}$$

By identification :

$\mathrm{a} = \mathrm{cos}(\omega_k.\alpha), b = sin(\omega_k.\alpha), c = -sin(\omega_k.\alpha), d = cos(\omega_k.\alpha)$

References

- Ashish Vaswaniet al. *Attention Is All You Need*, 12 Jun 2017
- Qiang Wang et al., *Learning Deep Transformer Models for Machine Translation*
- Bryan Lim et al., *Temporal Fusion Transformers for interpretable multi-horizon time series forecasting*
- Jay Alammar, The Illustrated Transformer
- Amirhossein Kazemnejad, Blog