

EARTH'S LAST STAND

Transformer

Mi AI Blog

A MICHAEL BAY FILM
TRANSFORMERS
DARK OF THE MOON

ĐƠN VỊ TỔ CHỨC



ĐƠN VỊ TÀI TRỢ



BIG OFF MÌ AI BLOG

08:30, 13.12.2020

Cafe Climax, Cung văn hoá Việt Xô
91 Trần Hưng Đạo, Hoàn Kiếm, Hà Nội

Miễn phí vào cửa | Các công nghệ update từ các công ty AI | Định hướng nghề nghiệp | Bốc thăm may mắn



Transformer là gì?

- Transformer là một mô hình học sâu được giới thiệu năm 2017, được dùng chủ yếu ở lĩnh vực xử lý ngôn ngữ tự nhiên (NLP).
- Có thể coi là SOTA - State Of The Art

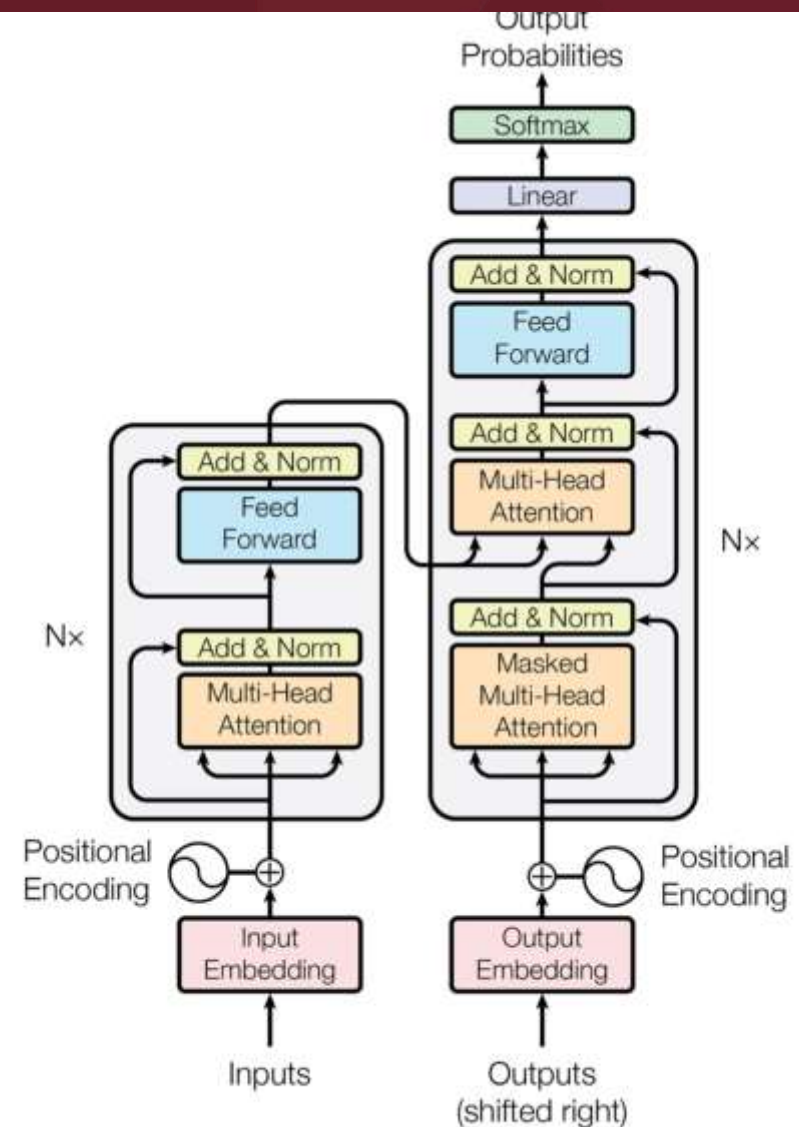
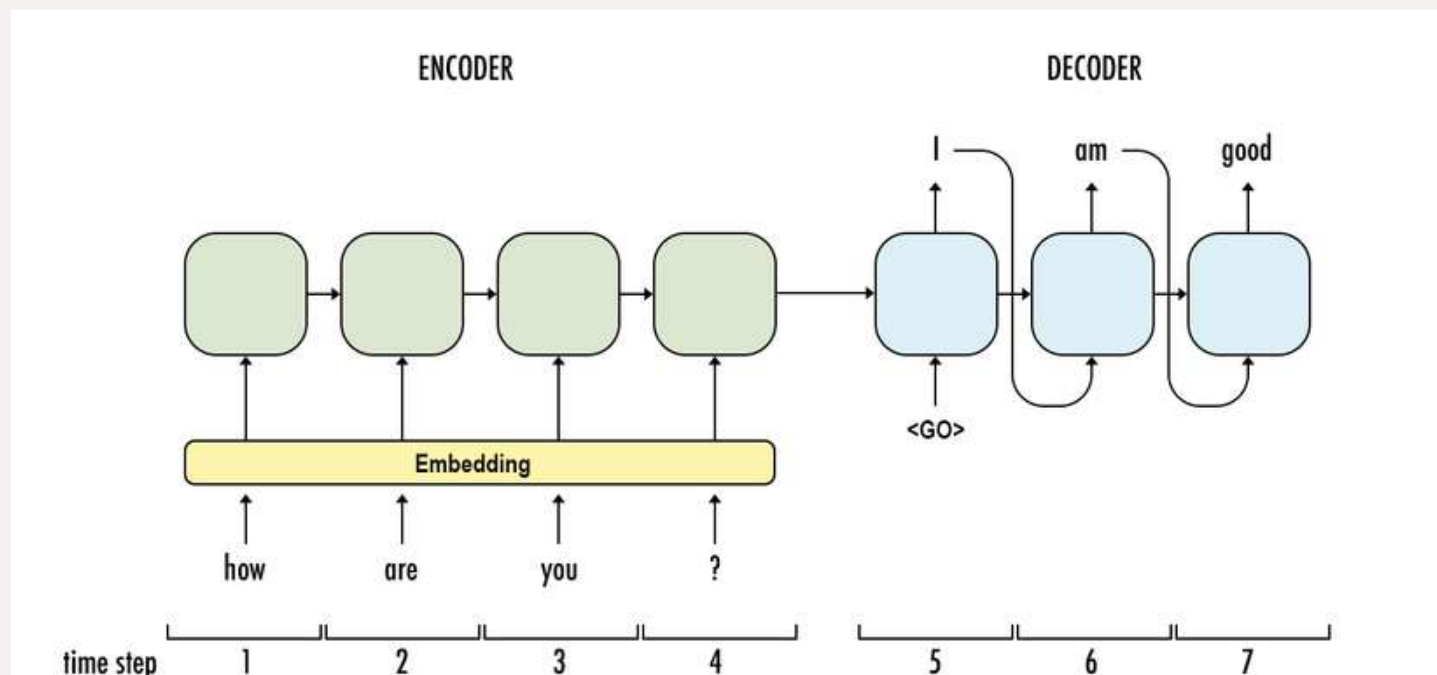


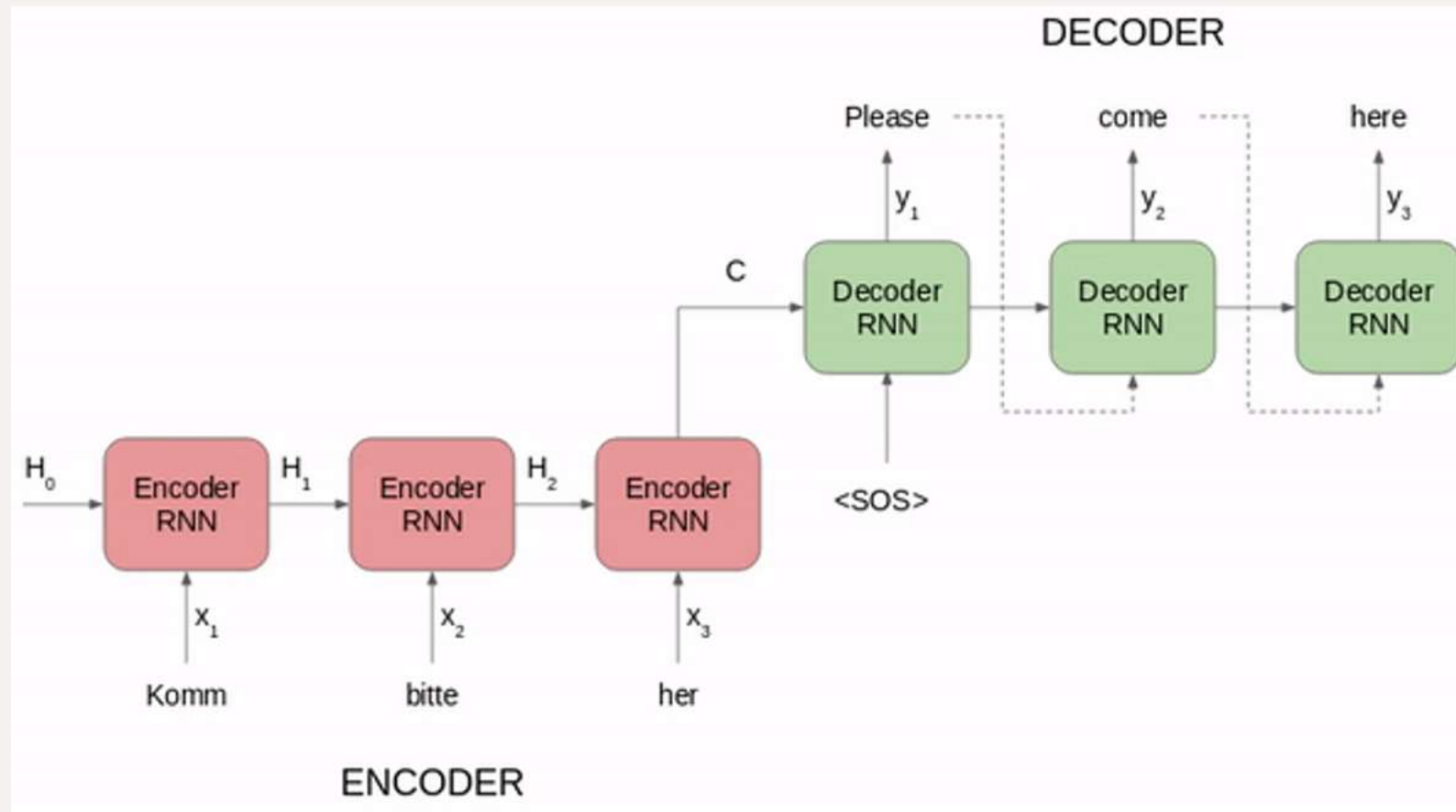
Figure 1: The Transformer - model architecture.

Ngày xưa ngày xưa (RNN)

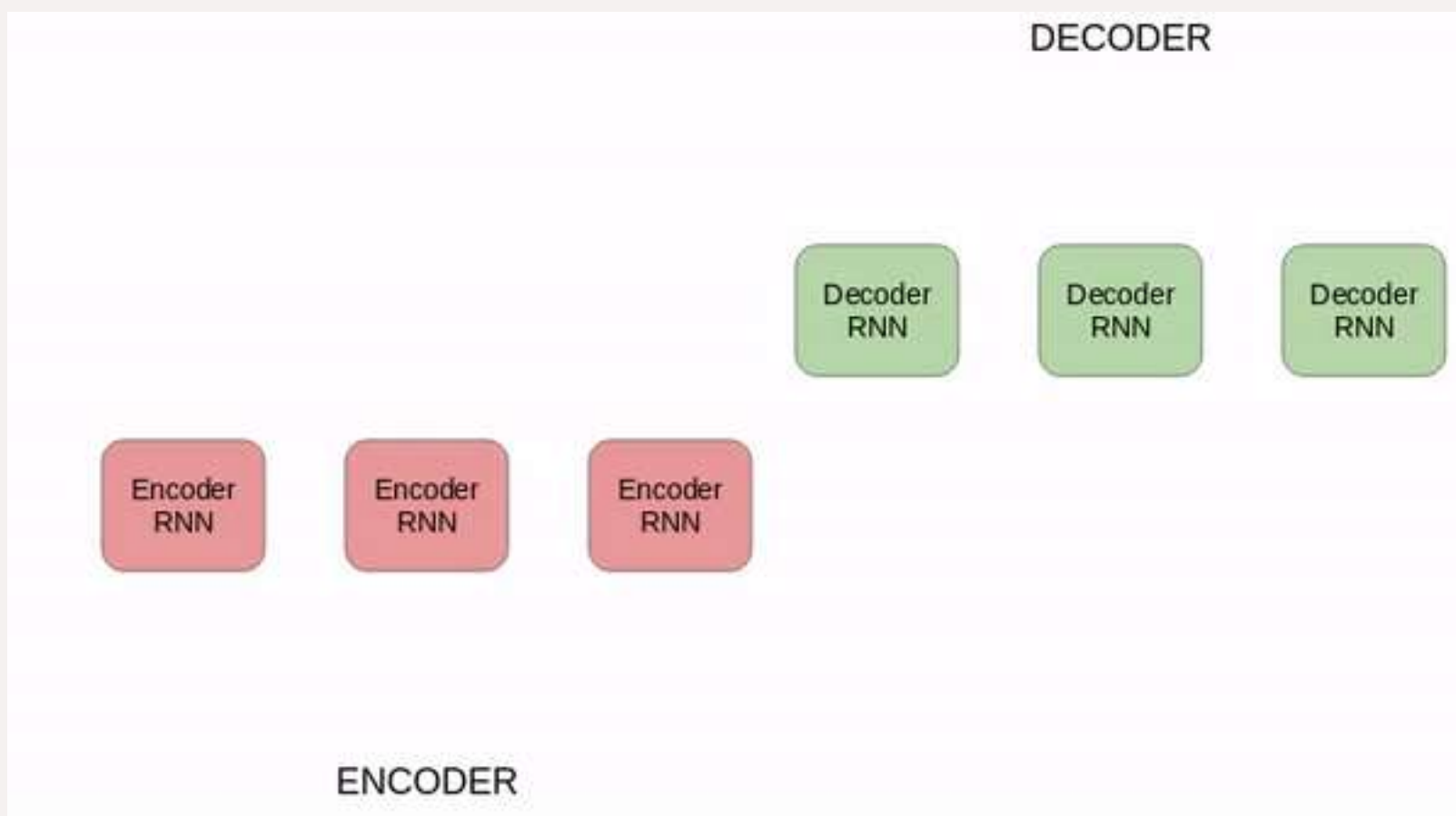
- Các bài toán Seq2Seq sử dụng kiến trúc mạng RNN, LSTM, GRU...
- Nhận input là một sequence và trả lại output cũng là một sequence. Ví dụ bài toán Q&A, input là câu hỏi "how are you ?" và output là câu trả lời "I am good". Phương pháp truyền thống sử dụng RNNs cho cả encoder (phần mã hóa input) và decoder (phần giải mã input và đưa ra output tương ứng)



Ngày xưa ngày xưa (RNN)

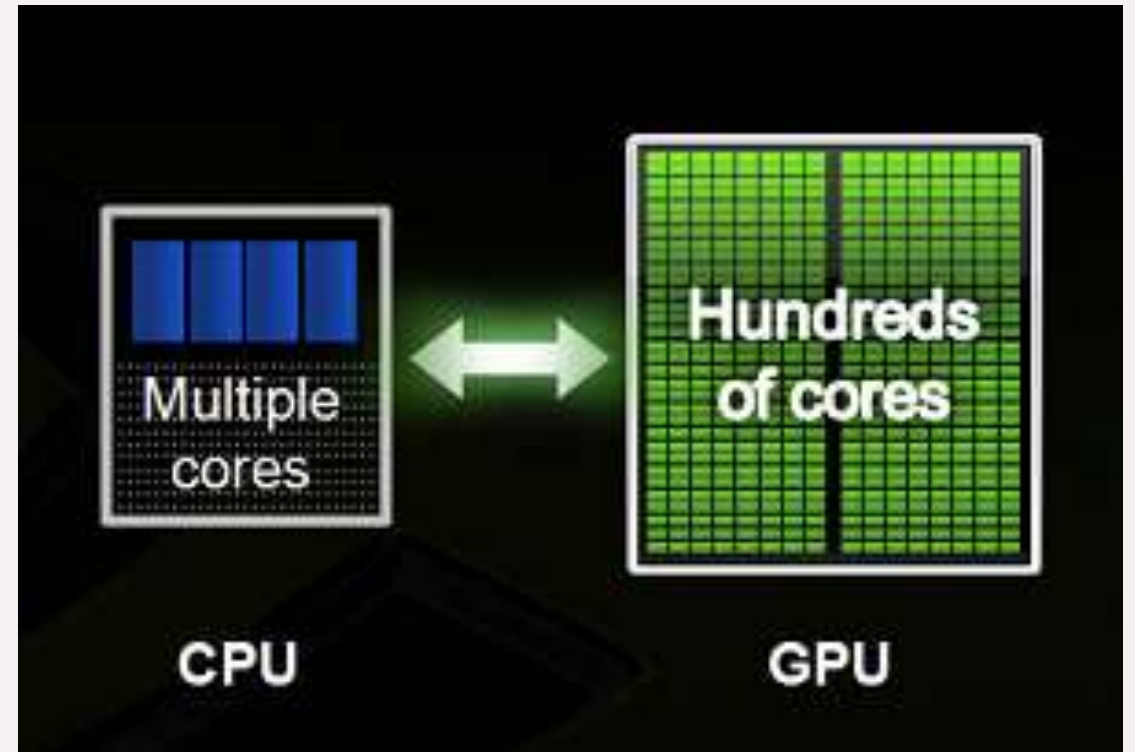


Ngày xưa ngày xưa (RNN)



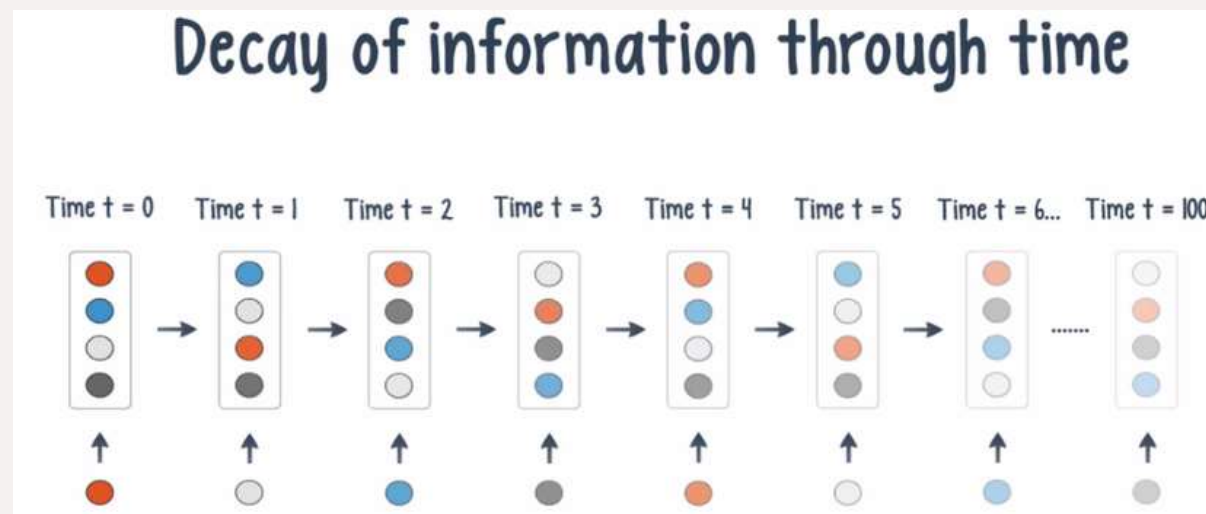
Có gì sai?

Không thể tính toán song song, train chậm và không tận dụng sức mạnh của GPU



Có gì sai?

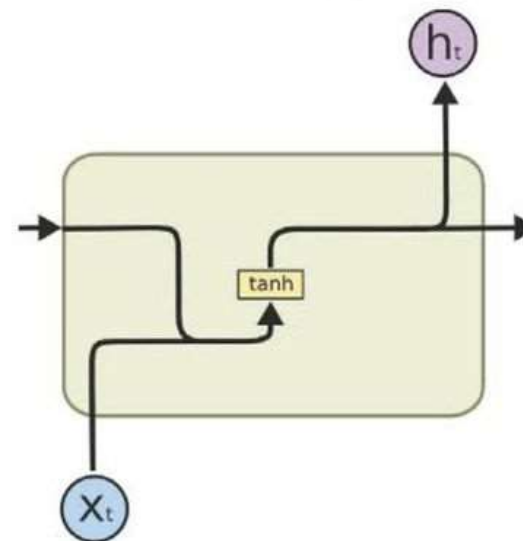
- Bị giảm thông tin, mỗi liên hệ qua các bước do triệt tiêu/bùng nổ Gradient. Đặc biệt với các câu dài.



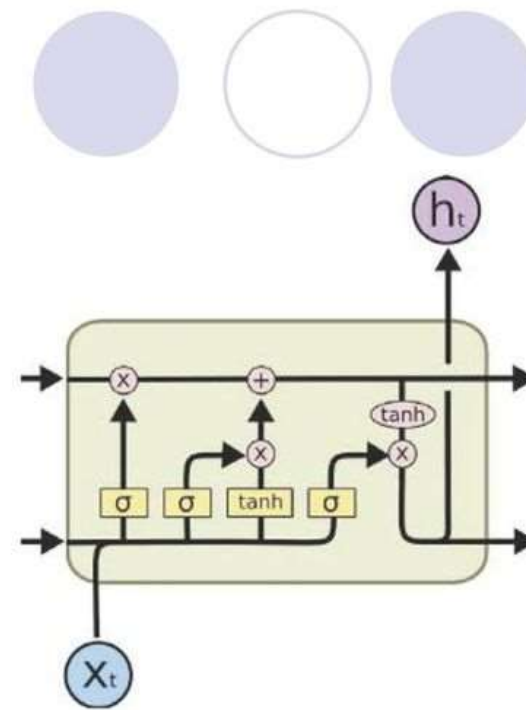
Có gì sai?

- LSTM có thể khắc phục đôi chút do có cổng cho thông tin đi qua giúp duy trì thông tin khi qua các step nhưng lại dẫn tới train chậm hơn nữa do LSTM khá phức tạp

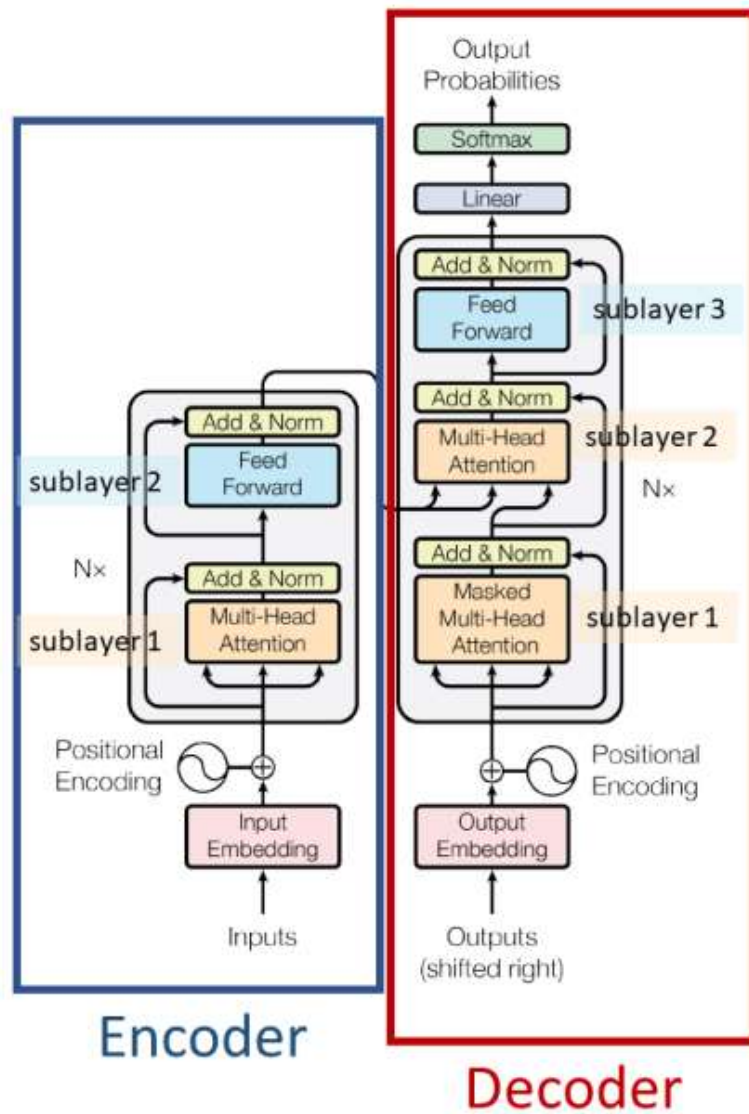
RNN vs LSTM



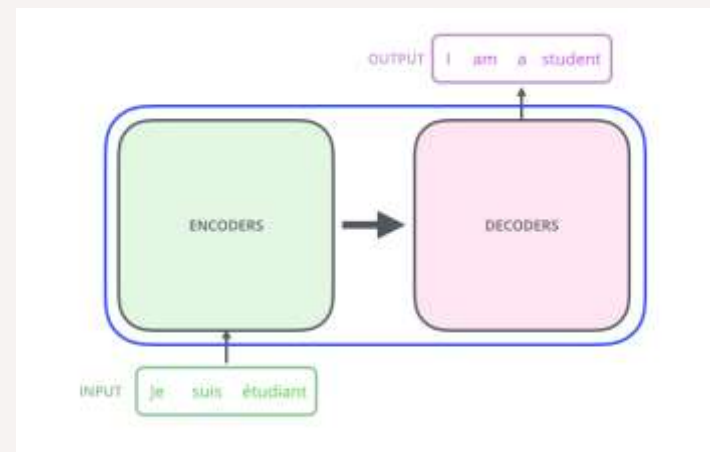
(a) RNN



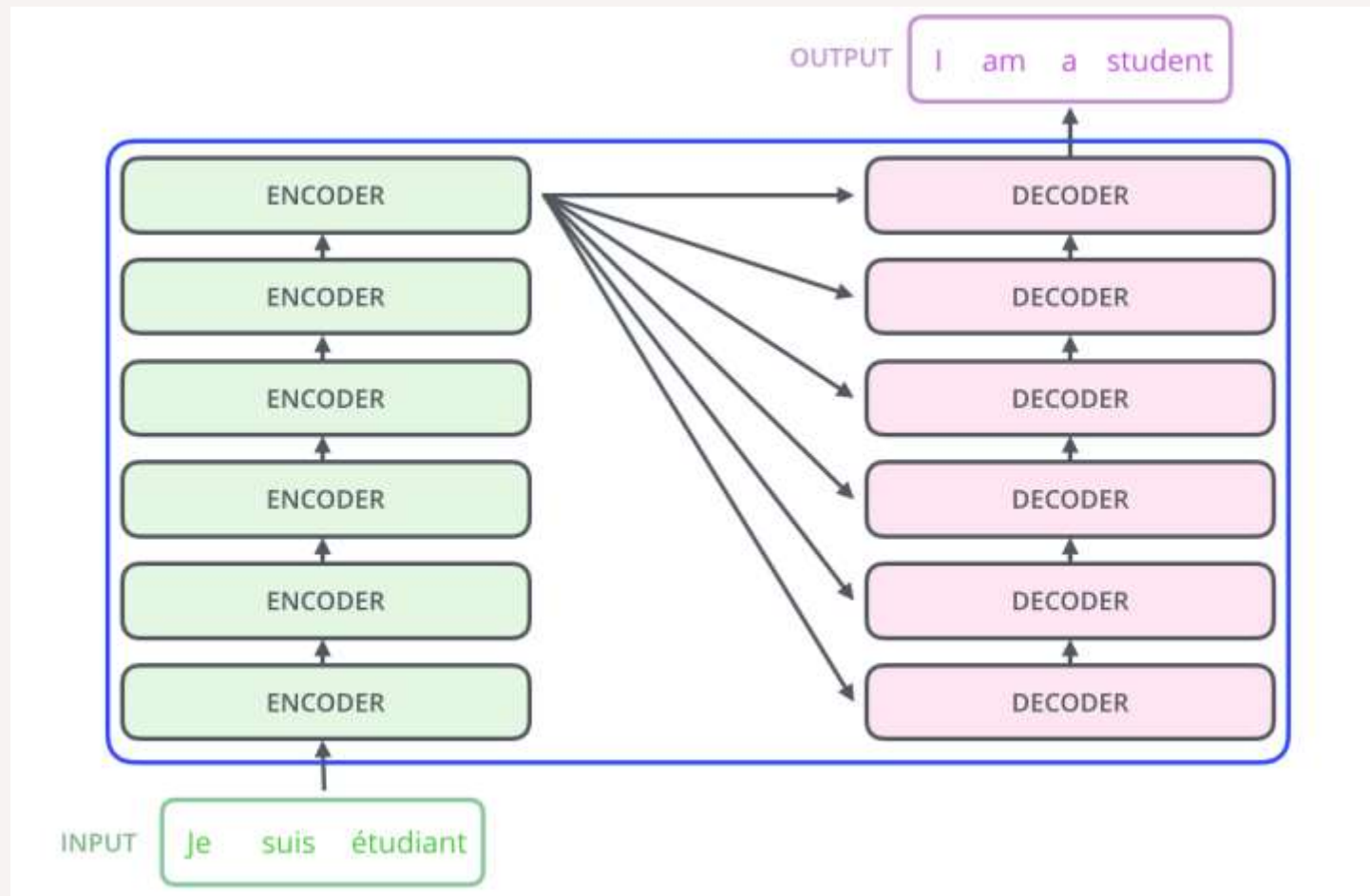
(b) LSTM



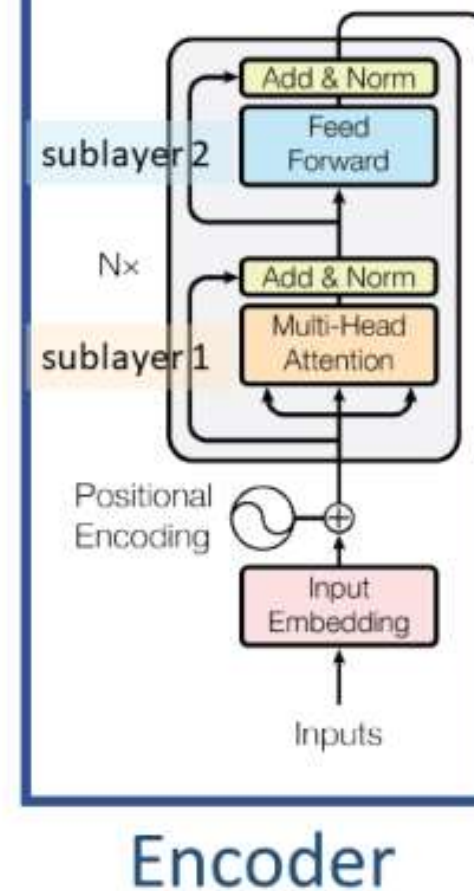
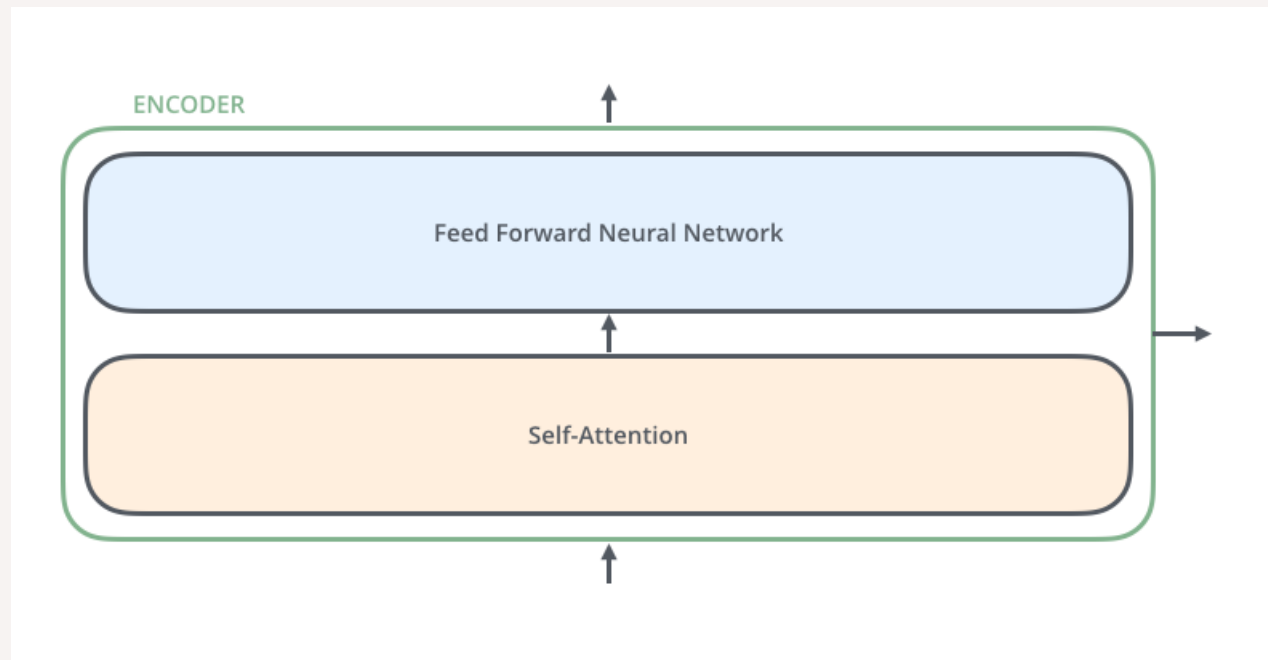
Và đó là sứ
mệnh của
Transformer

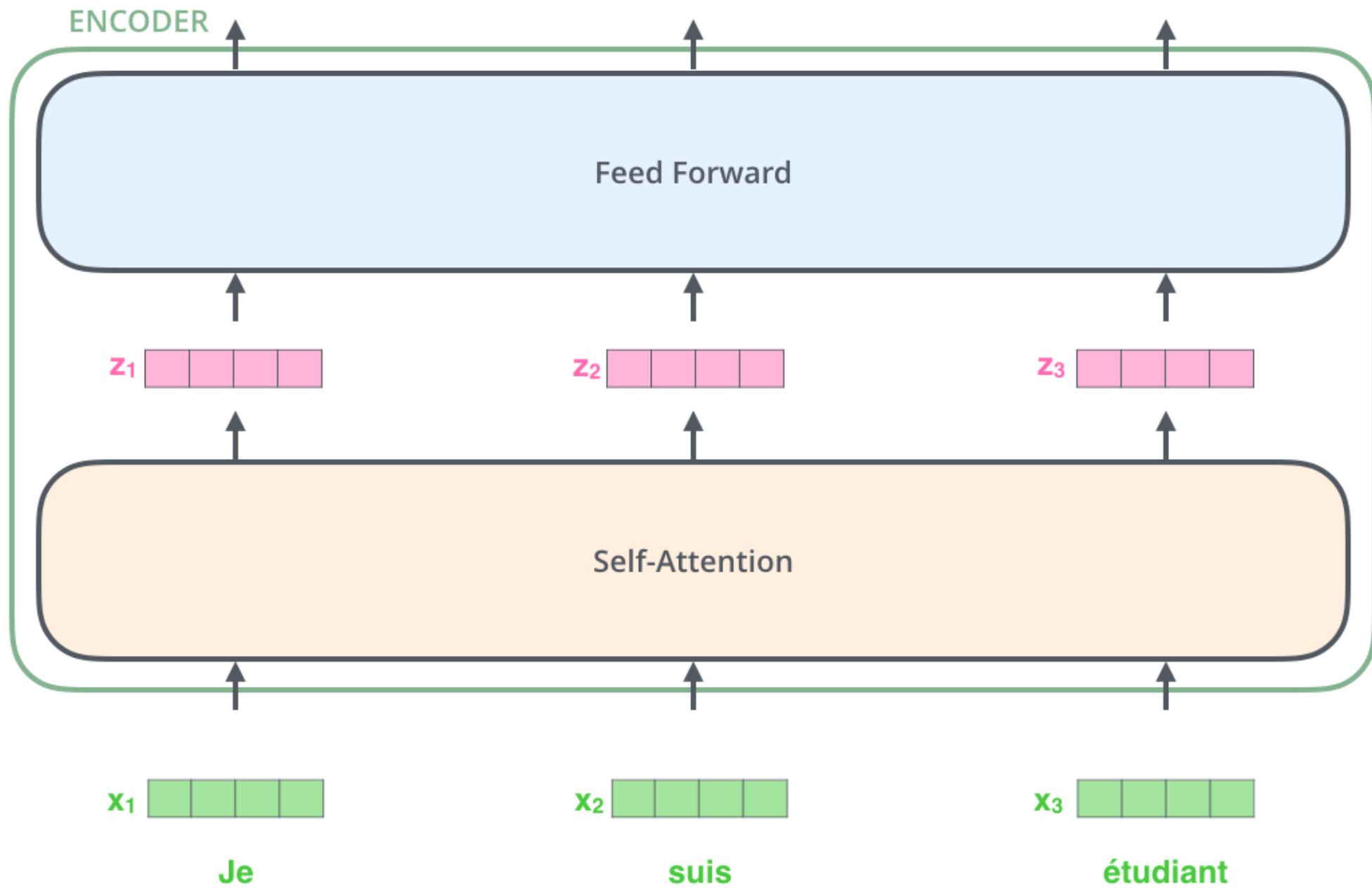


Kiến trúc



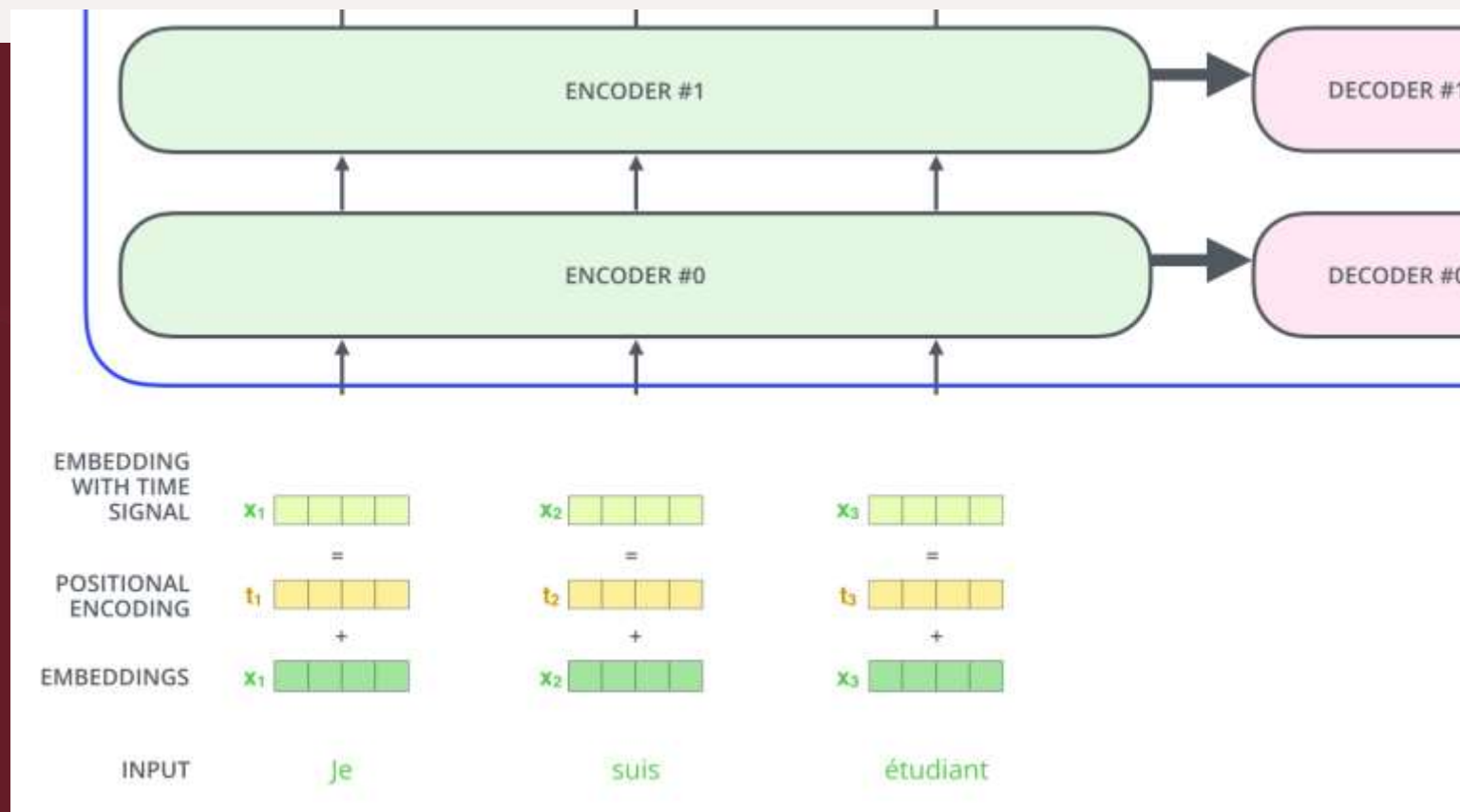
Mở xẻ ông Encoder





Positional Encoding

- Tất cả các vector từ phi vào mạng cùng 1 lúc, song song
- Cần một cơ chế để "note" lại vị trí các từ trong câu
- Vector PE tham gia cuộc chơi (size = word embedding)



Positional Encoding

- Có thể dùng index but...
- Tác giả dùng hàm sin, cosin

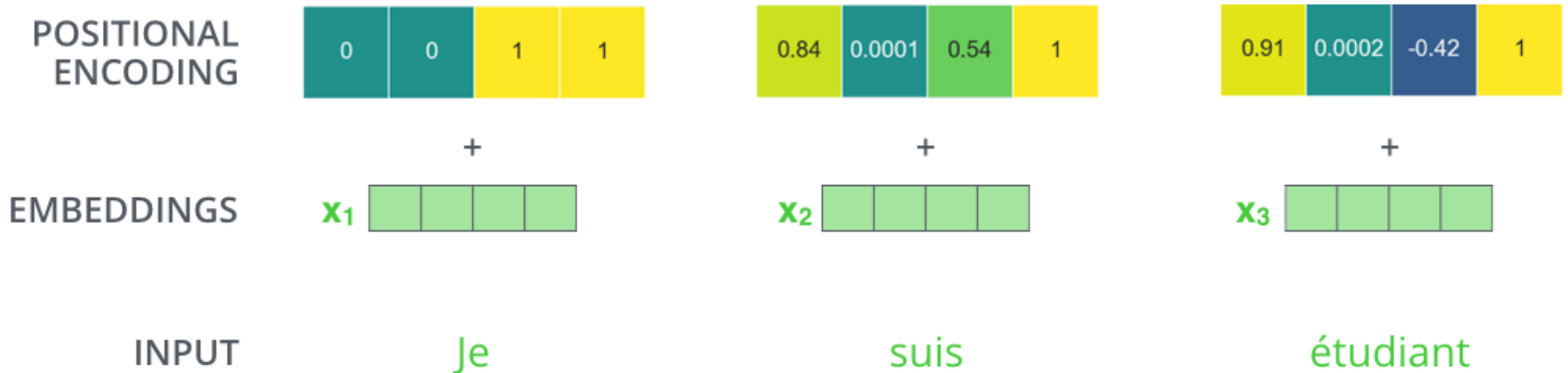
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

$$p_{i,j} = \begin{cases} \sin\left(\frac{i}{10000^{\frac{j}{d_{emb-dim}}}}\right) & \text{if } j \text{ is even} \\ \cos\left(\frac{i}{10000^{\frac{j-1}{d_{emb-dim}}}}\right) & \text{if } j \text{ is odd} \end{cases}$$

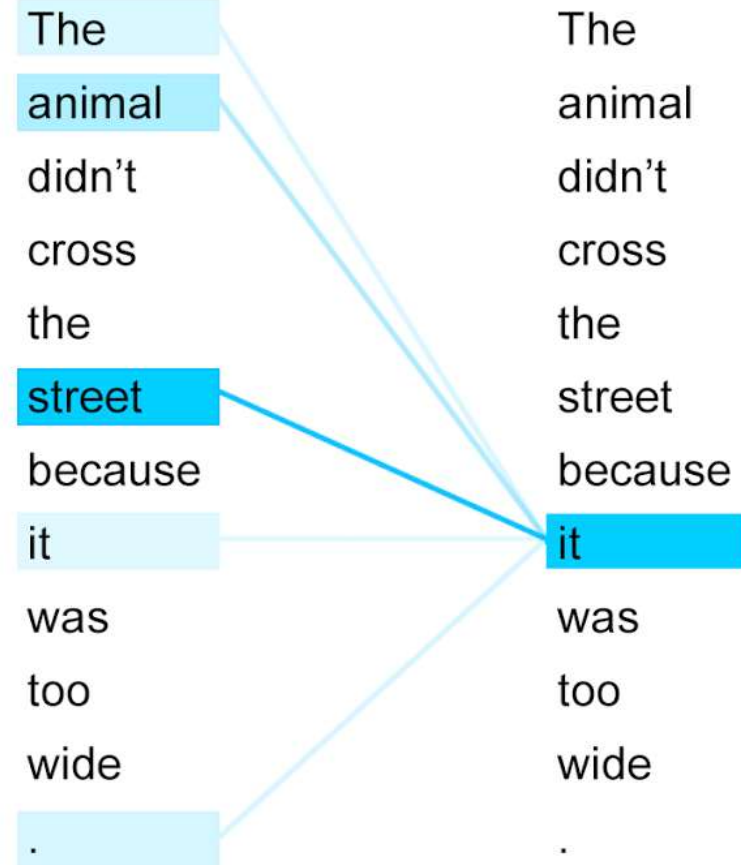
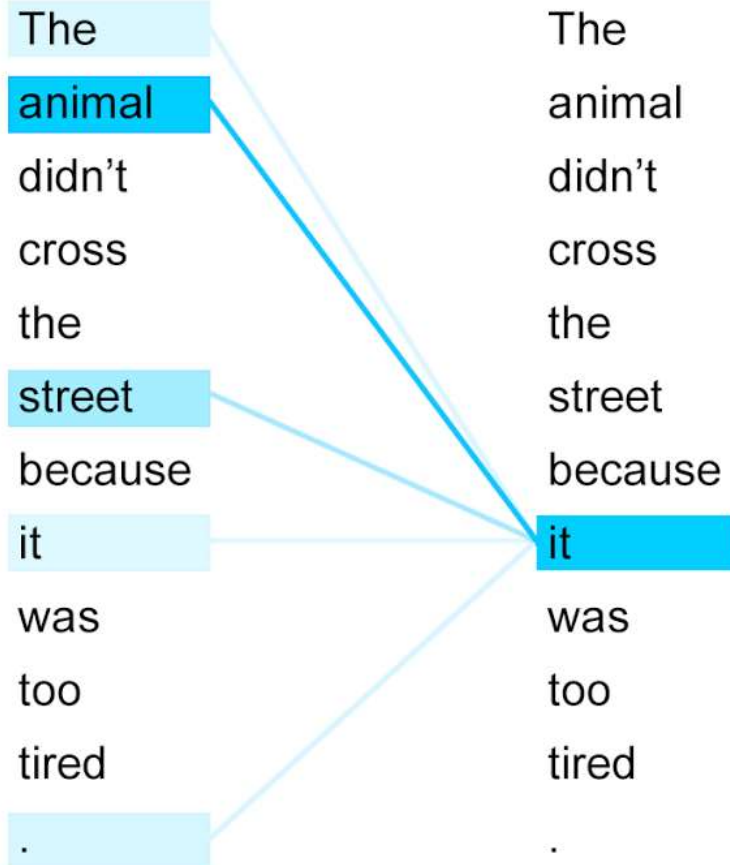
Positional Encoding

- Có thể dùng index but...
- Tác giả dùng hàm sin, cosin

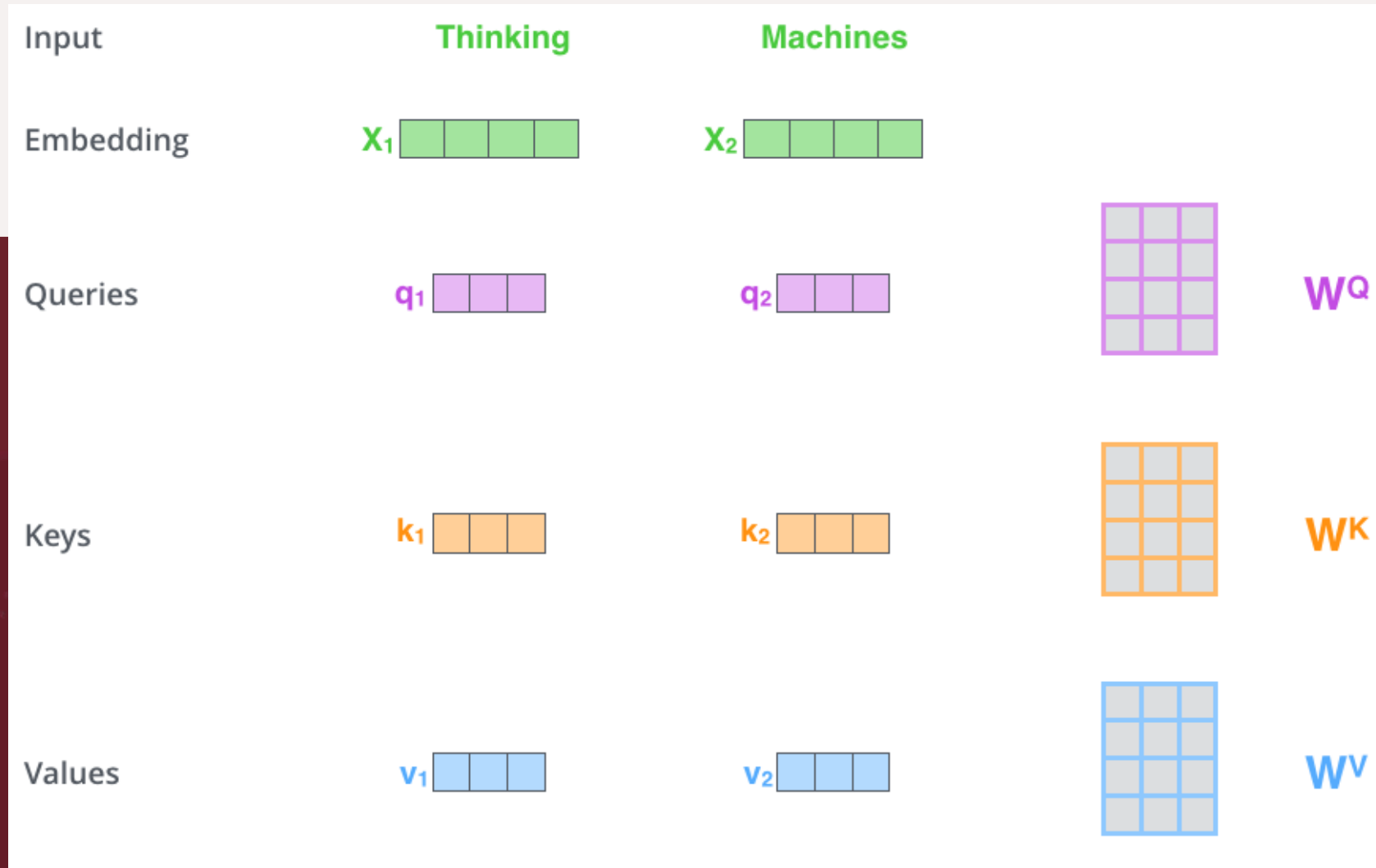


Self Attention

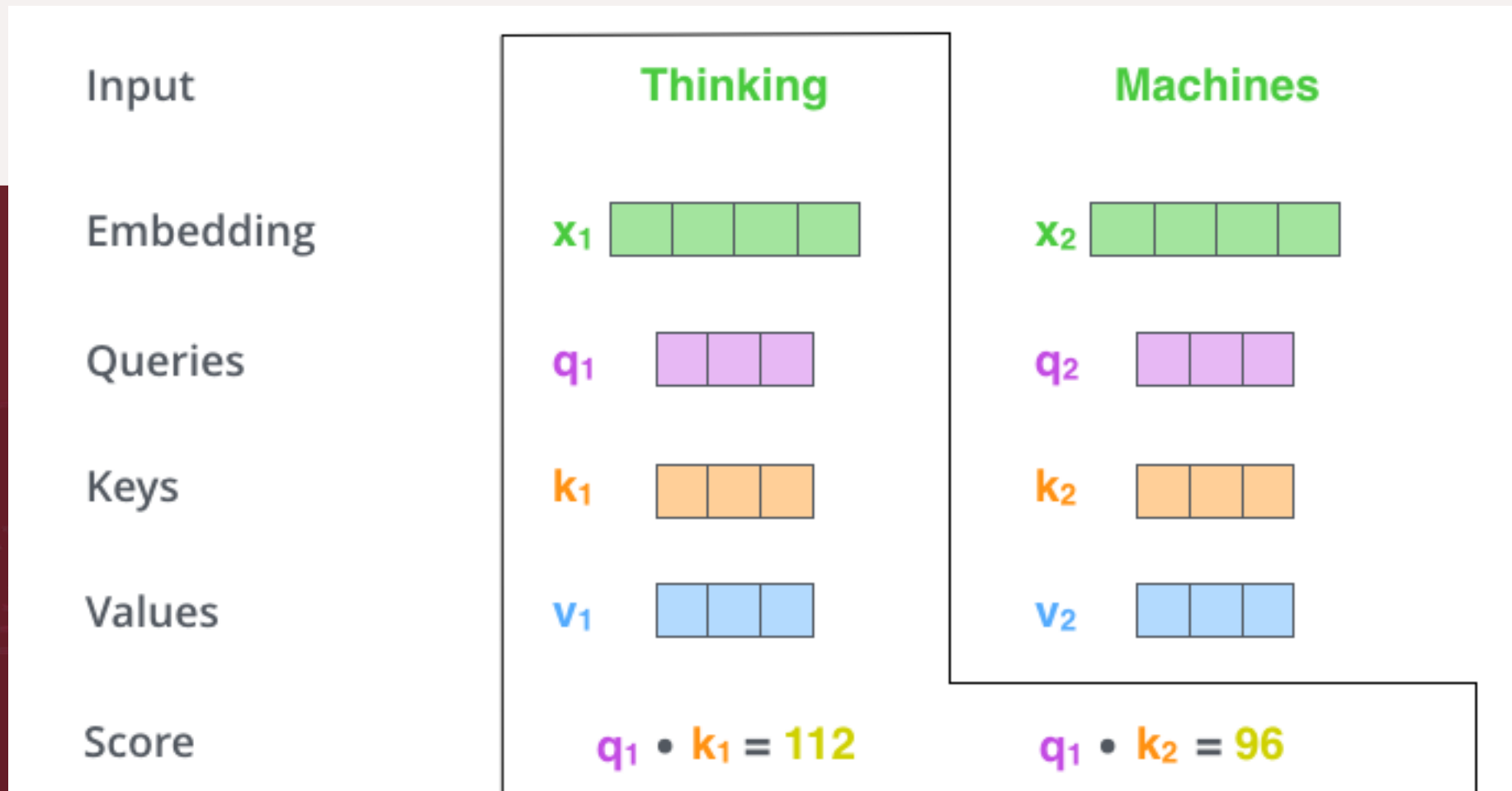
- Tạo ra quan hệ giữa các từ trong câu
- Khi được mã hoá (encode) nó sẽ mang thêm thông tin của các từ liên quan



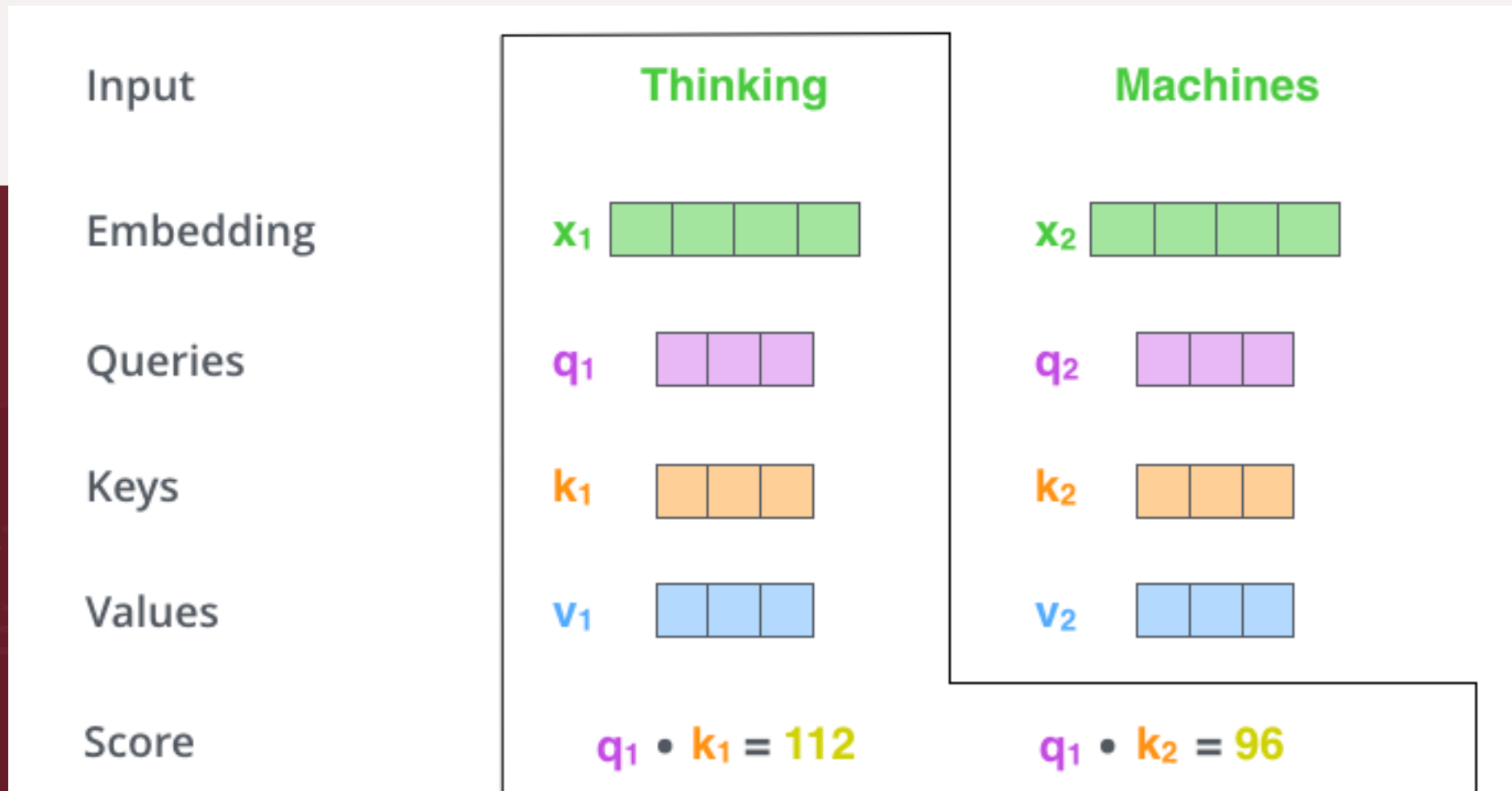
Self Attention



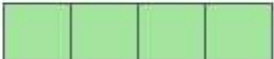
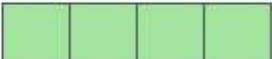

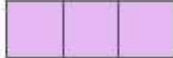

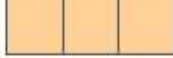

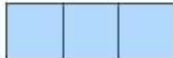
Self Attention



Self Attention



Self Attention

Input	Thinking	Machines
Embedding	x_1 	x_2 
Queries	q_1 	q_2 
Keys	k_1 	k_2 
Values	v_1 	v_2 
Score	$q_1 \cdot k_1 = 112$	$q_1 \cdot k_2 = 96$
Divide by 8 ($\sqrt{d_k}$)	14	12
Softmax	0.88	0.12

Self Attention

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ($\sqrt{d_k}$)

Softmax

Softmax

X
Value

Sum

Thinking

x_1 

q_1 

k_1 

v_1 

$$q_1 \cdot k_1 = 112$$

14

0.88

v_1 

z_1 

Machines

x_2 

q_2 

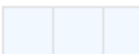
k_2 

v_2 

$$q_1 \cdot k_2 = 96$$

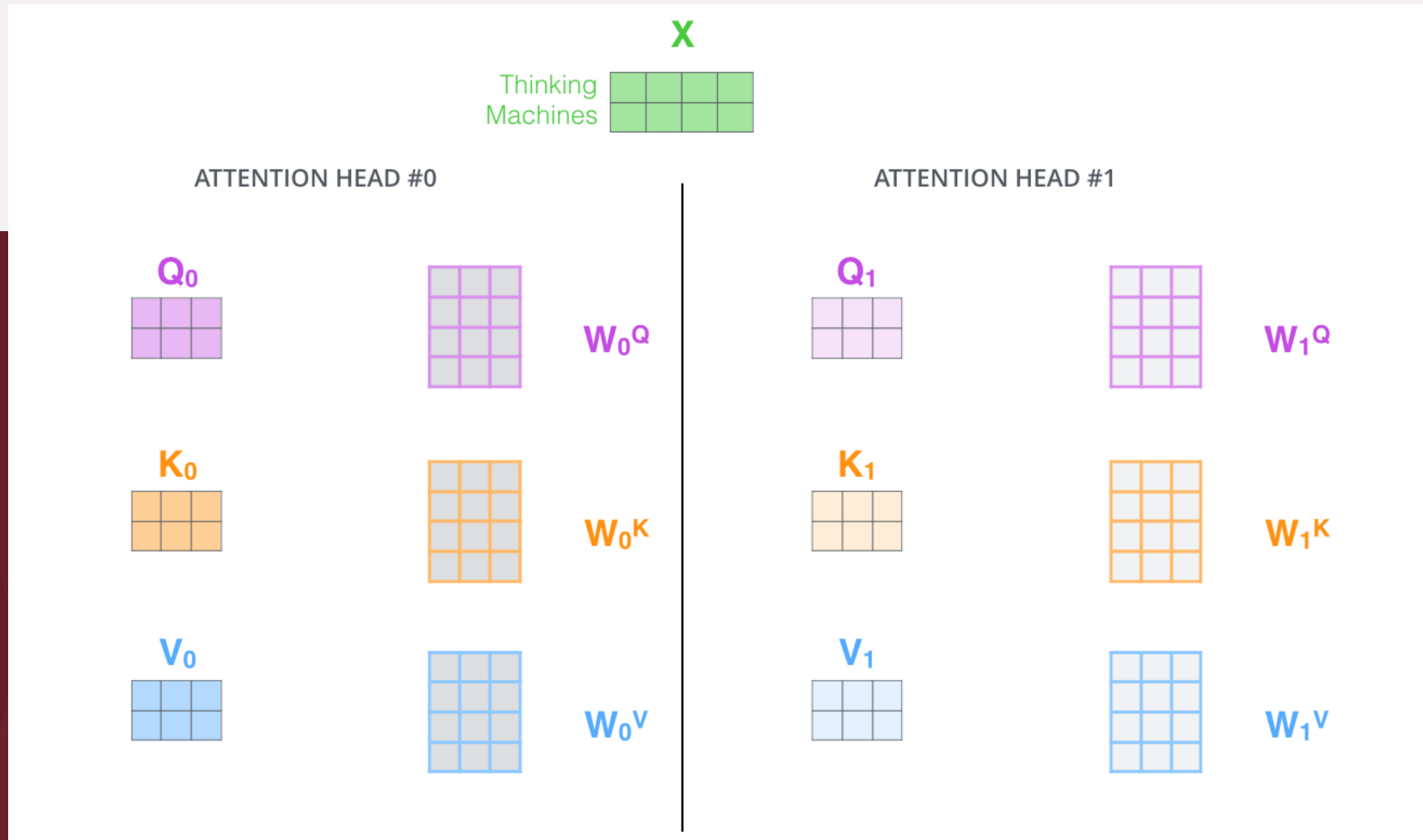
12

0.12

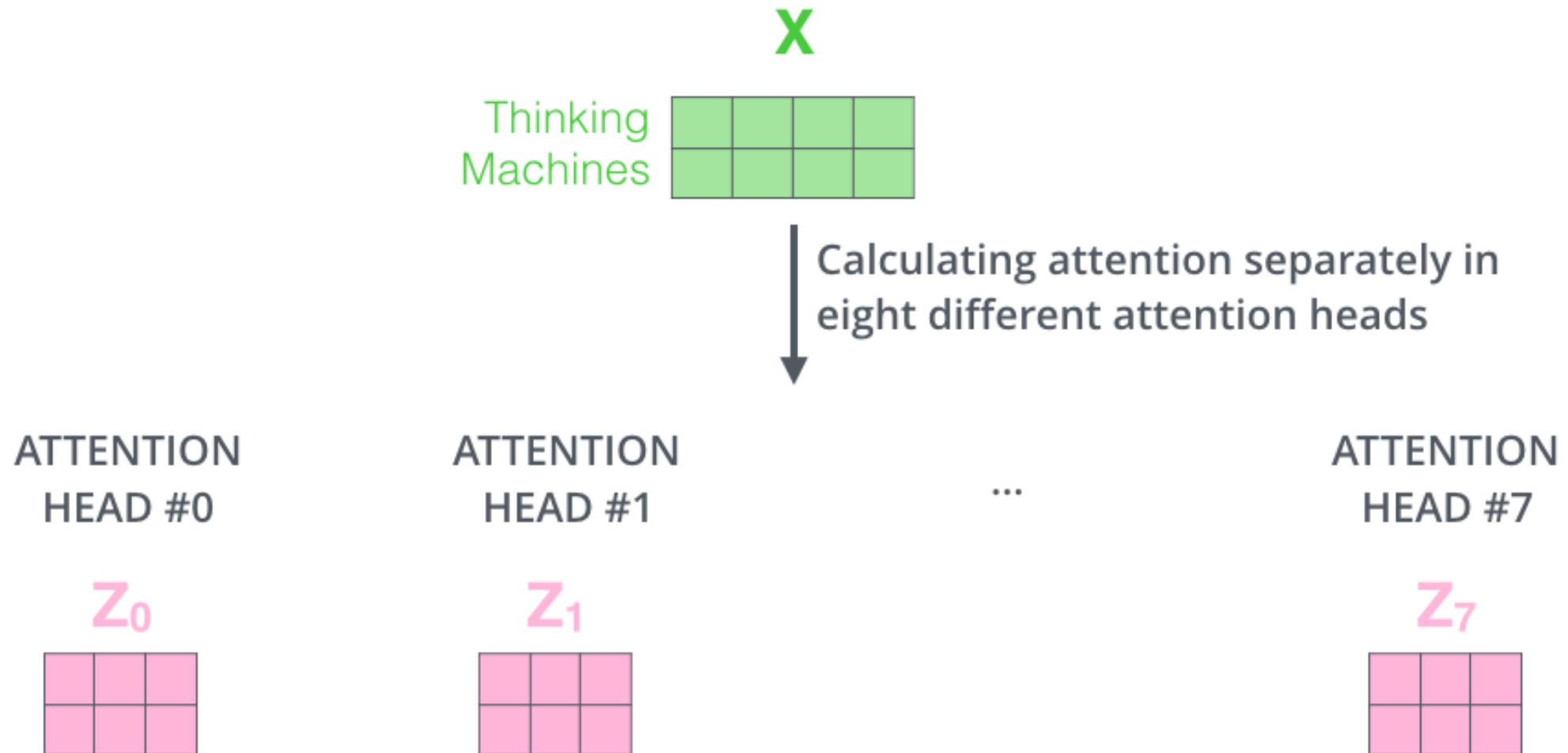
v_2 

z_2 

Multi-head

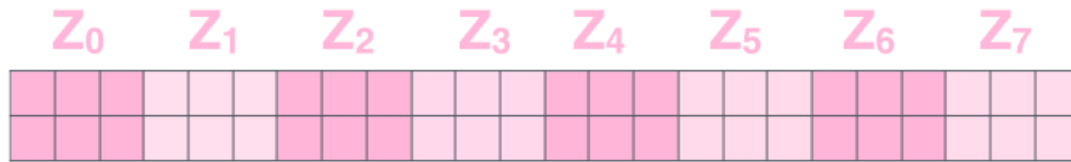


Multi-head



Multi-head

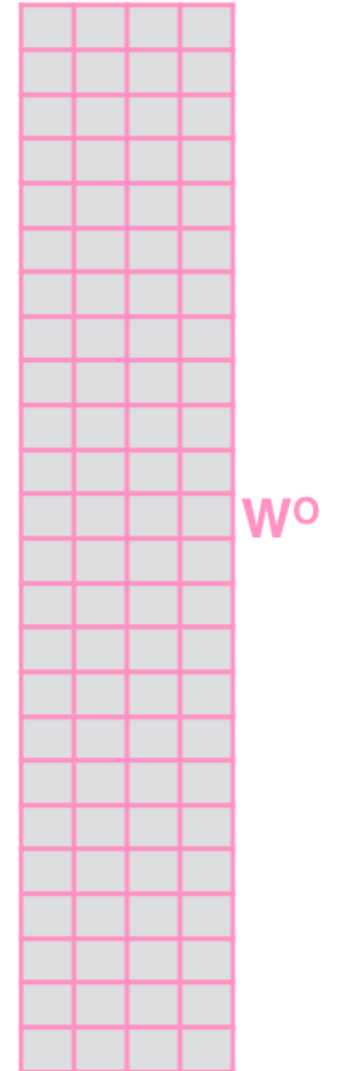
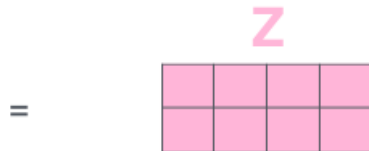
1) Concatenate all the attention heads



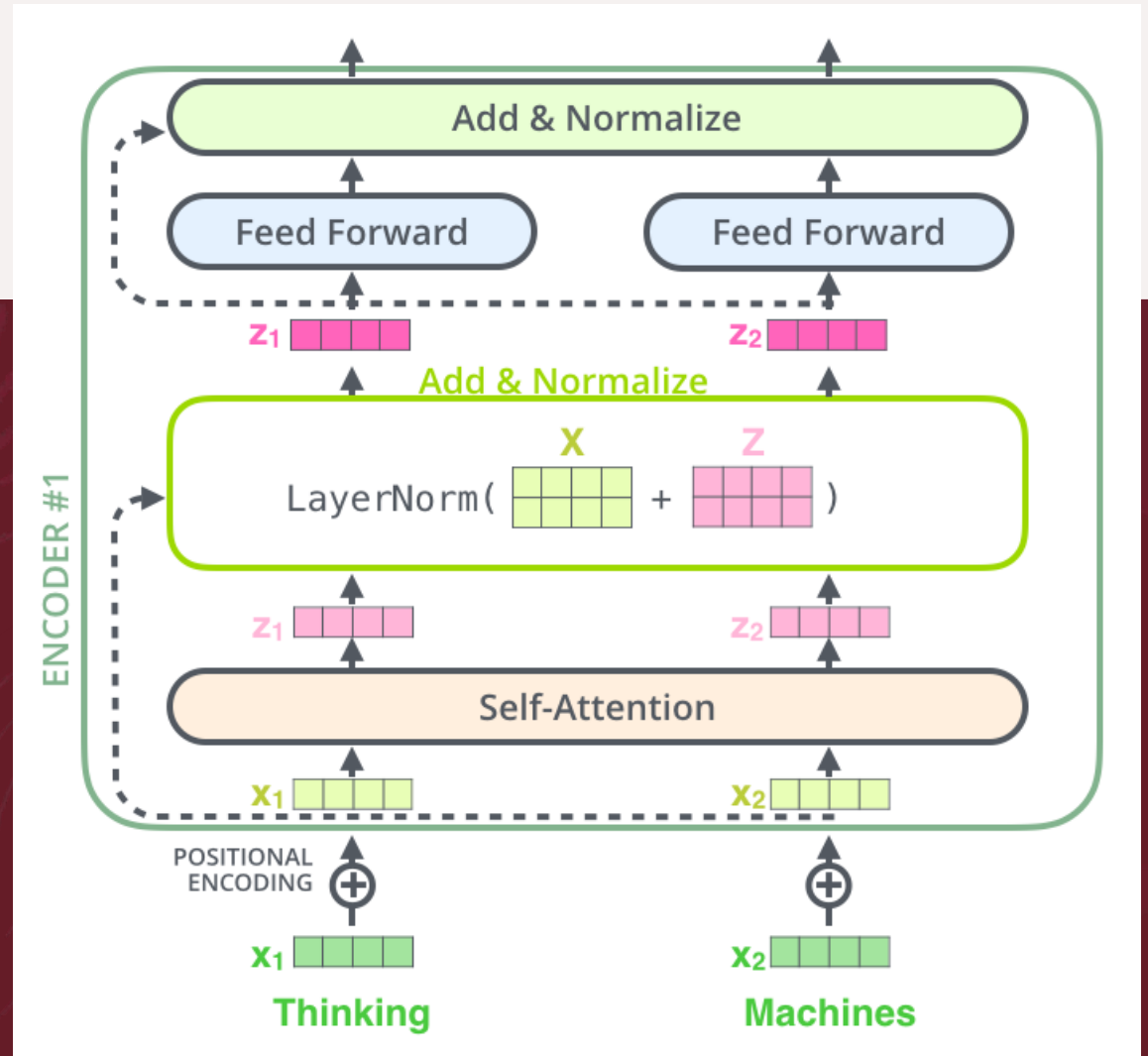
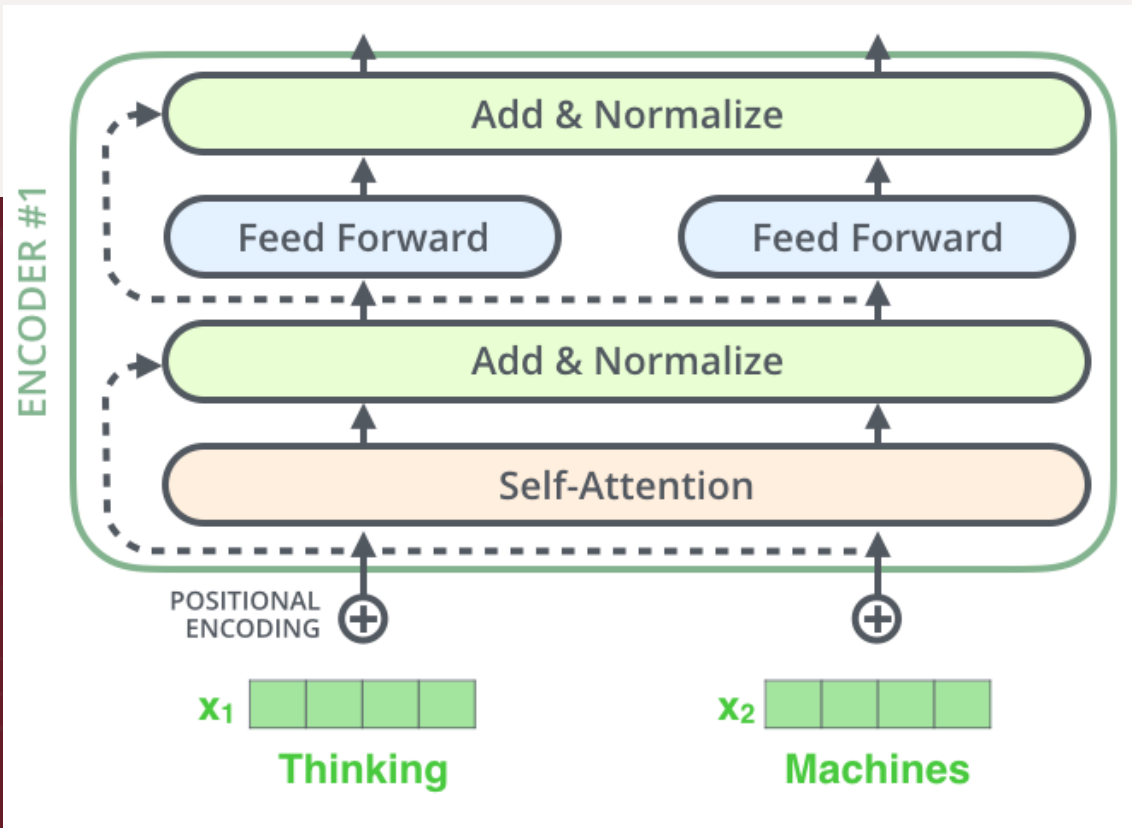
2) Multiply with a weight matrix W^O that was trained jointly with the model

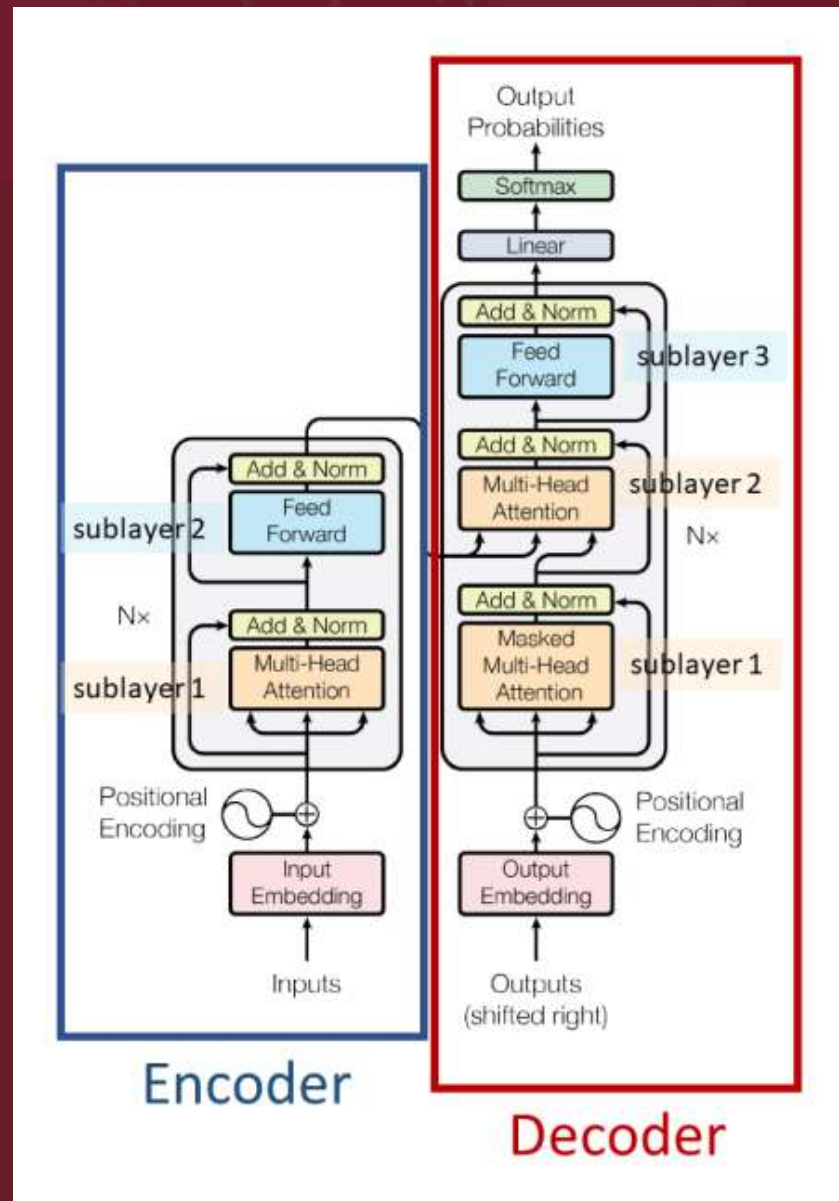
\times

3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN



The residual





Và ông Decoder

Hi,
how
are
you?



Transformers
Decoder

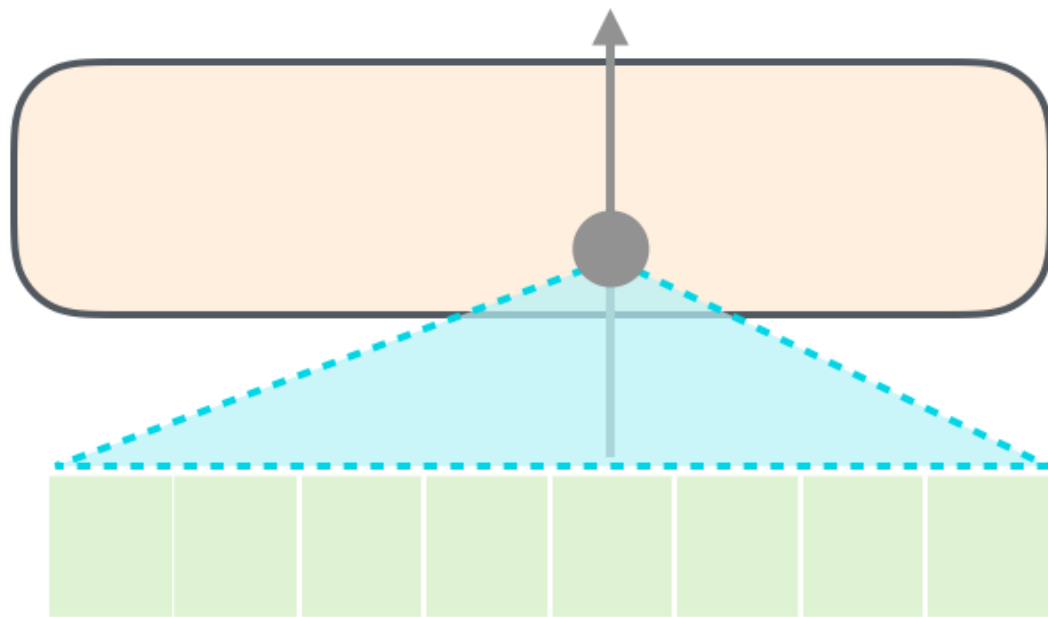


<start>

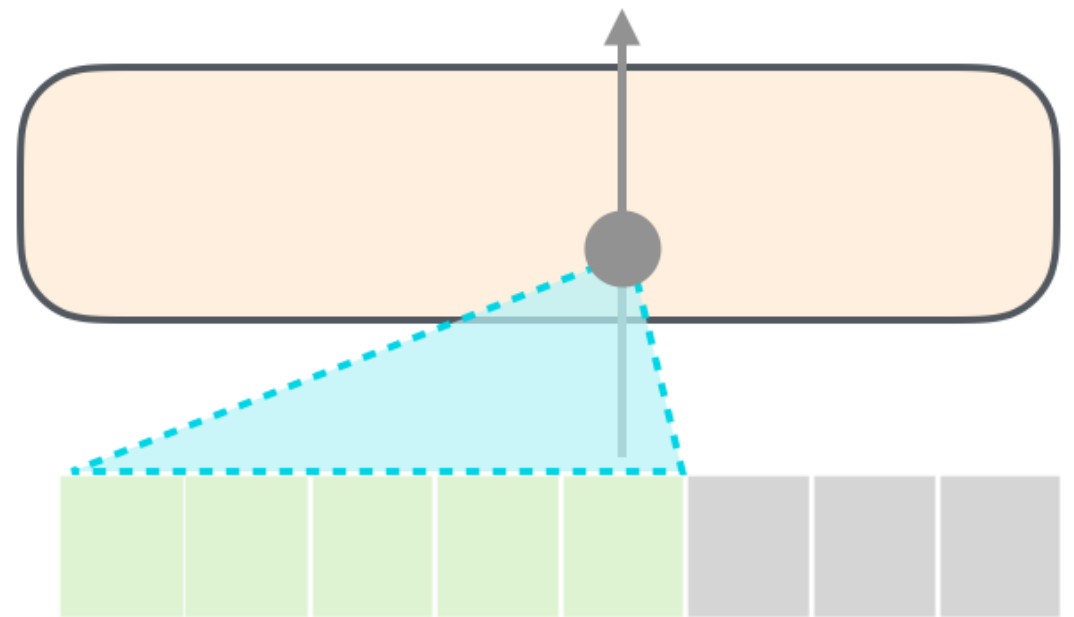
Và ông Decoder

Masked Attention

Self-Attention



Masked Self-Attention



Masked Attention

Scaled Scores

0.7	0.1	0.1	0.1
0.1	0.6	0.2	0.1
0.1	0.3	0.6	0.1
0.1	0.3	0.3	0.3

+

Look-Ahead Mask

0	-inf	-inf	-inf
0	0	-inf	-inf
0	0	0	-inf
0	0	0	0

=

Masked Scores

0.7	-inf	-inf	-inf
0.1	0.6	-inf	-inf
0.1	0.3	0.6	-inf
0.1	0.3	0.3	0.3

Masked Attention

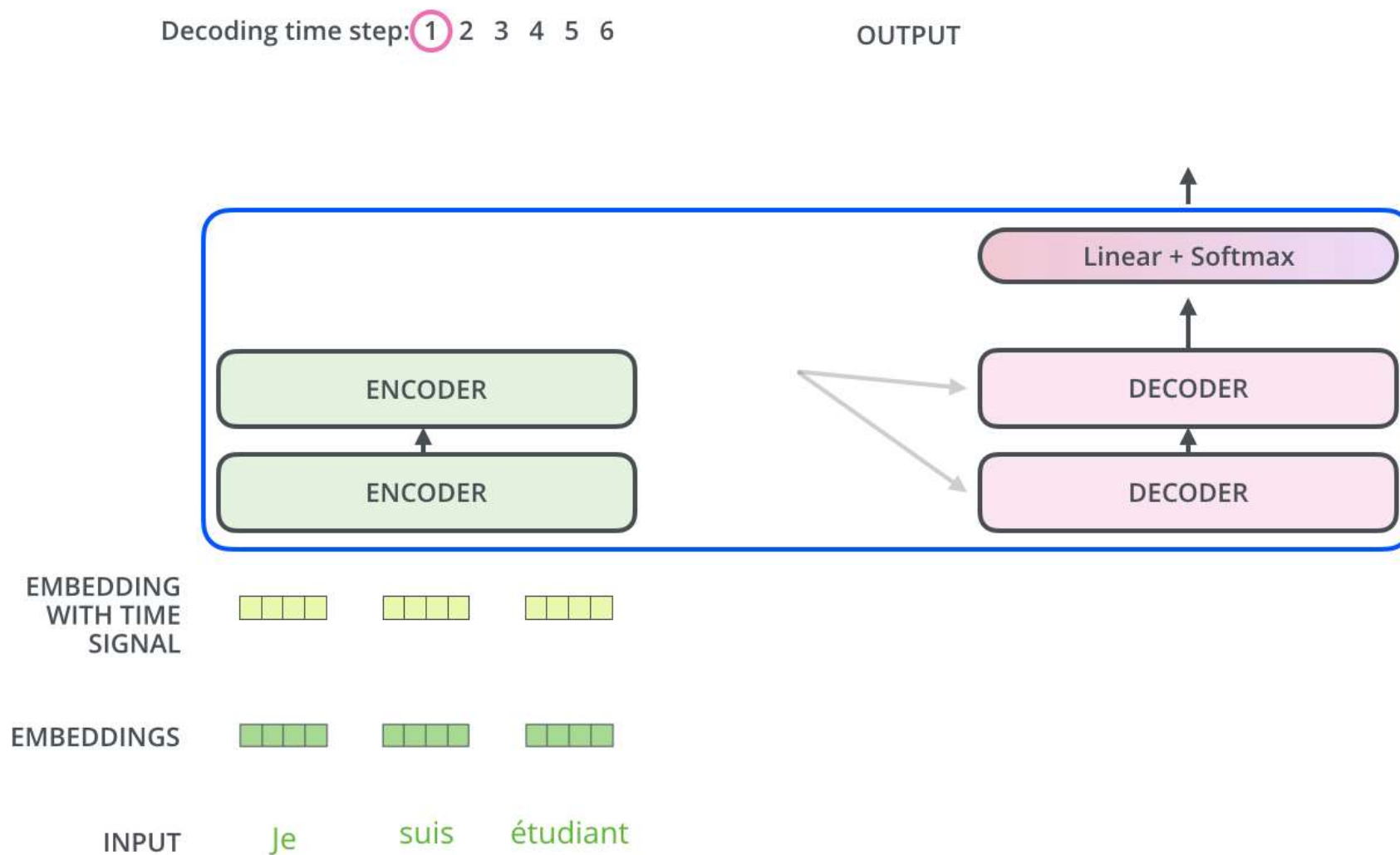
Softmax(

0.7	-inf	-inf	-inf
0.1	0.6	-inf	-inf
0.1	0.3	0.6	-inf
0.1	0.3	0.3	0.3

) =

	<start>	I	am	fine
<start>	1	0	0	0
I	0.37	0.62	0	0
am	0.26	0.31	0.43	0
fine	0.21	0.26	0.26	0.26

Encode-Decode Attention

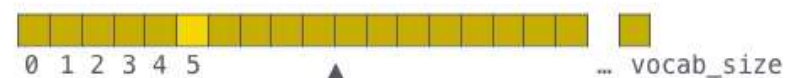


Linear + Softmax

Which word in our vocabulary
is associated with this index?

Get the index of the cell
with the highest value
(**argmax**)

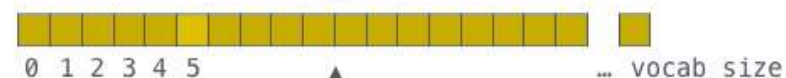
log_probs



am

5

logits



Softmax

Linear

Decoder stack output



Decoding time step: 1 2 3 4 5 6

OUTPUT |

