TRANSFORMER : Attention Is All You Need

엔자이너 연구실

2022-03-13

Contents

- RNN
- LSTM
- GRU
- Transformer : Attention Is All You Need

How to implement GNNs in my study

- Attention Is All You Need

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

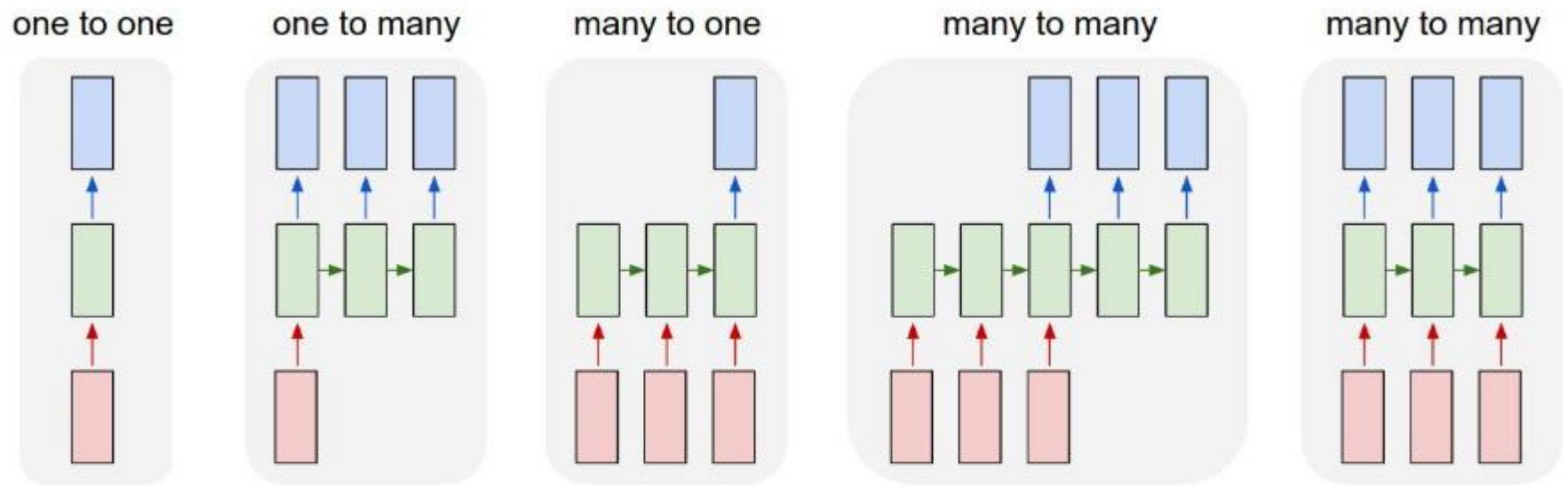
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

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.

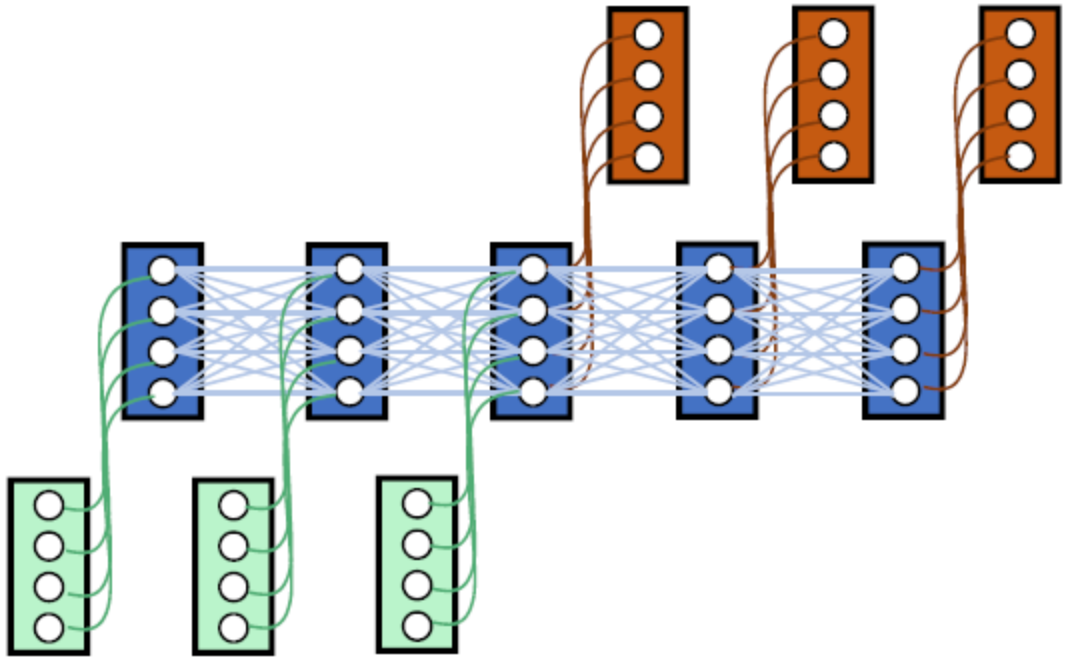
RNN

- Many to Many



RNN

- Many to Many



RNN

- Word Embedding

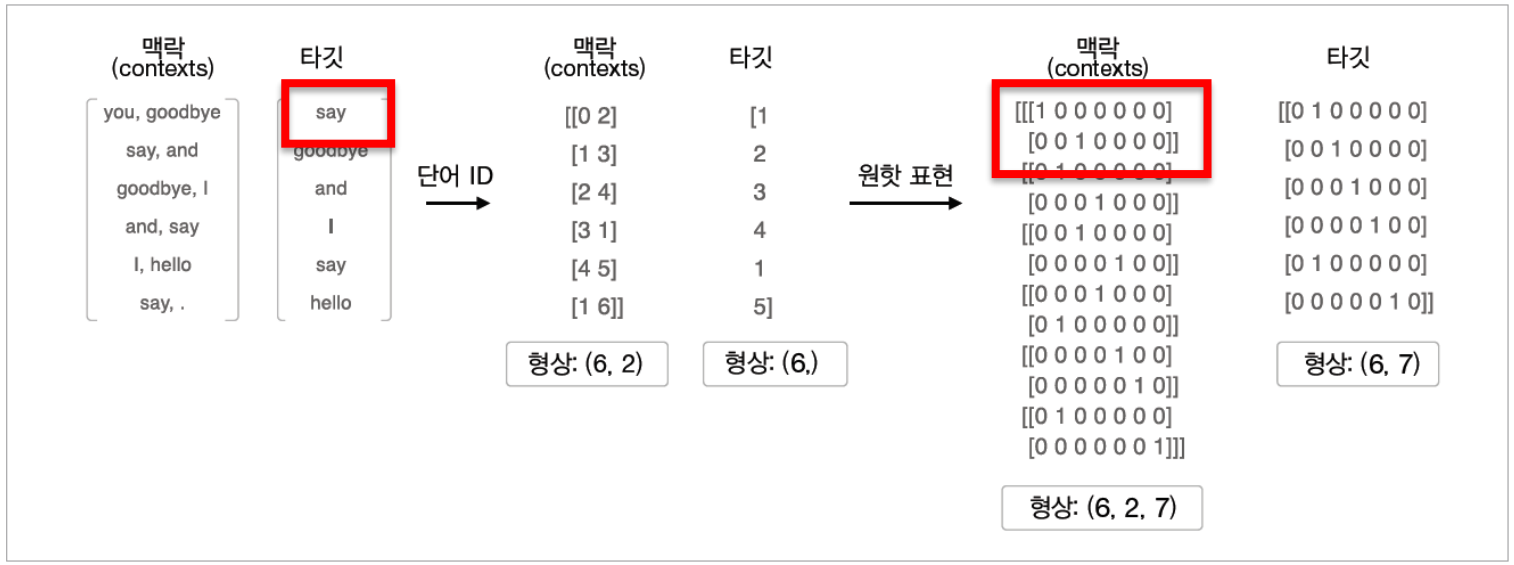
그림 3-16 말뭉치에서 맥락과 타깃을 만드는 예

말뭉치	맥락(contexts)	타깃
you say goodbye and I say hello .	you, goodbye	say
you say goodbye and I say hello .	say, and	goodbye
you say goodbye and I say hello .	goodbye, I	and
you say goodbye and I say hello .	and, say	I
you say goodbye and I say hello .	I, hello	say
you say goodbye and I say hello .	say, .	hello

RNN

- Word Embedding
  - You say goodbye and I say hello

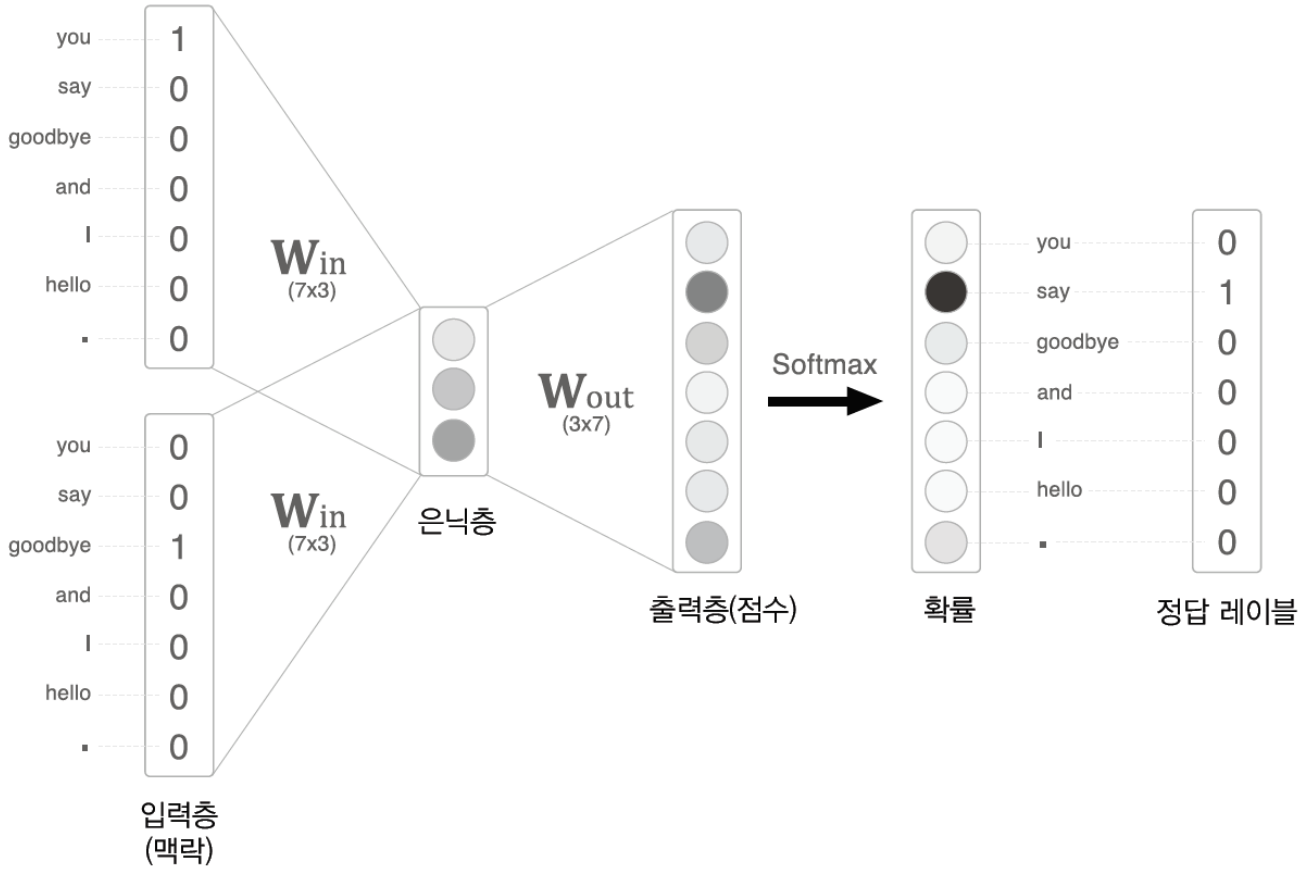
그림 3-18 '맥락'과 '타깃'을 원핫 표현으로 변환하는 예



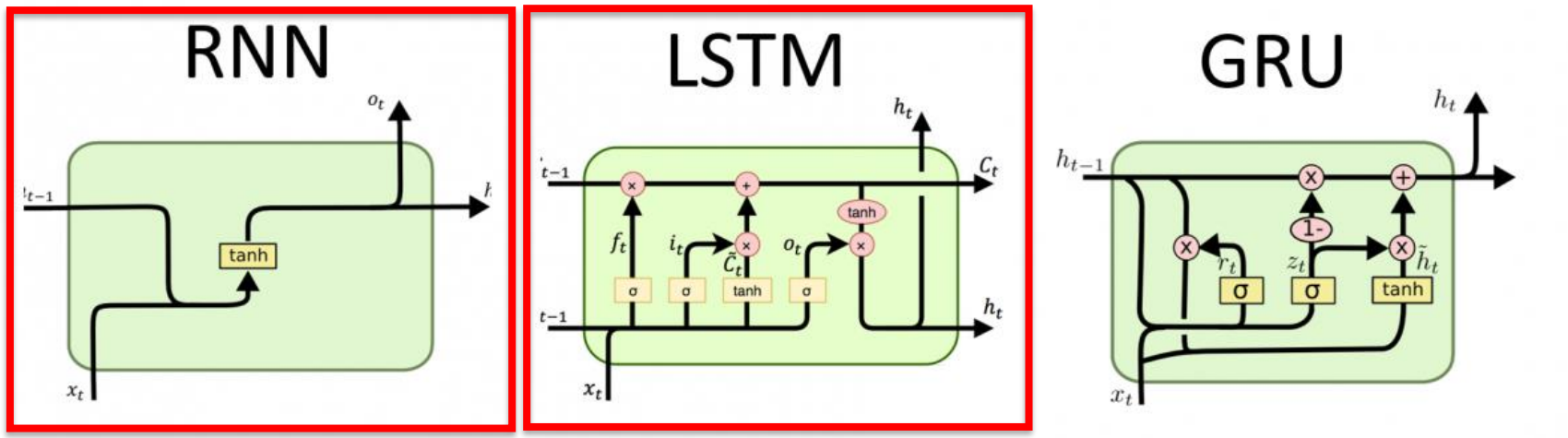
RNN

- Word Embedding

그림 3-12 CBOW 모델의 구체적인 예(노드 값의 크기를 흑백의 진하기로 나타냄)

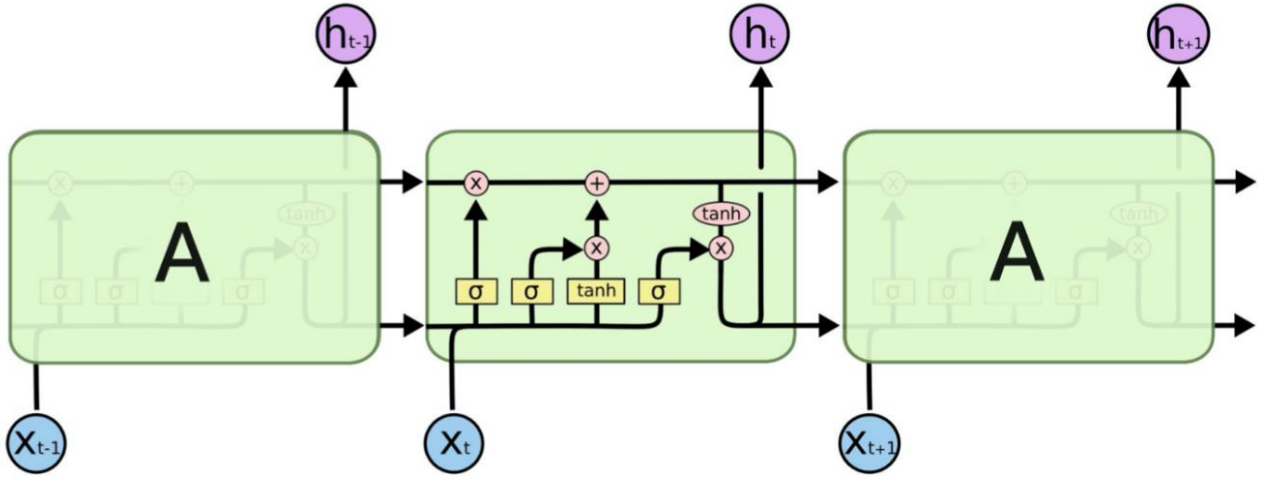


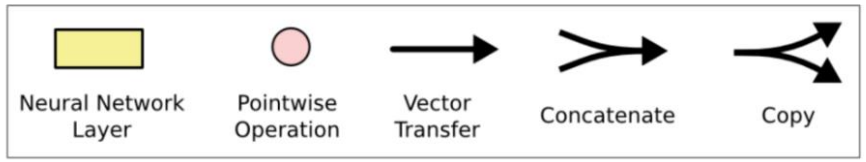
RNN



LSTM

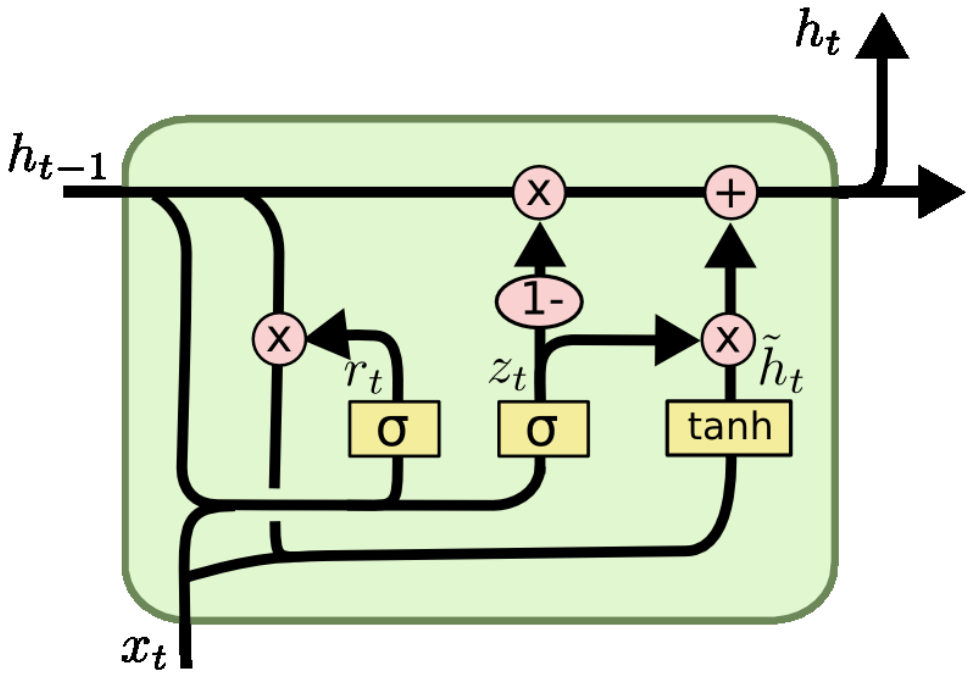
- Architecture



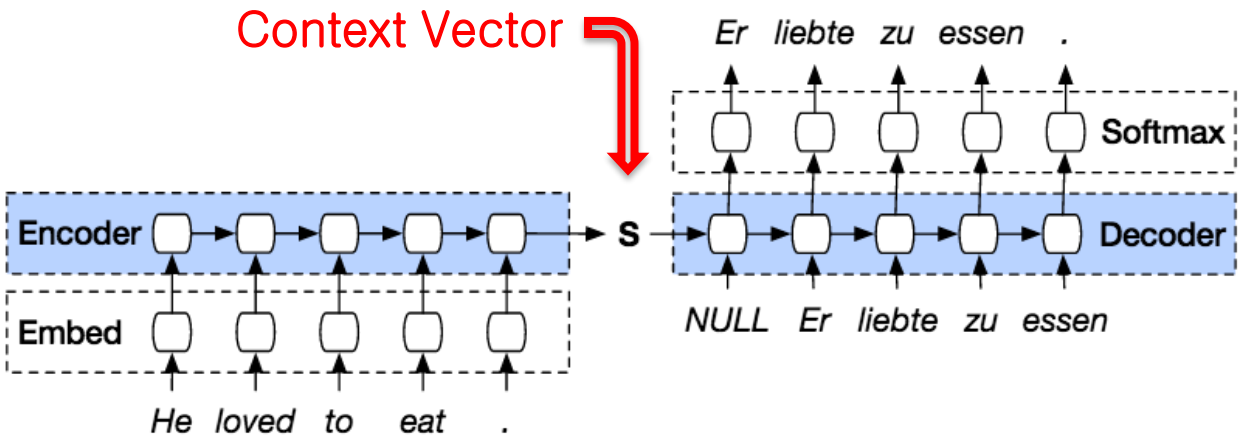


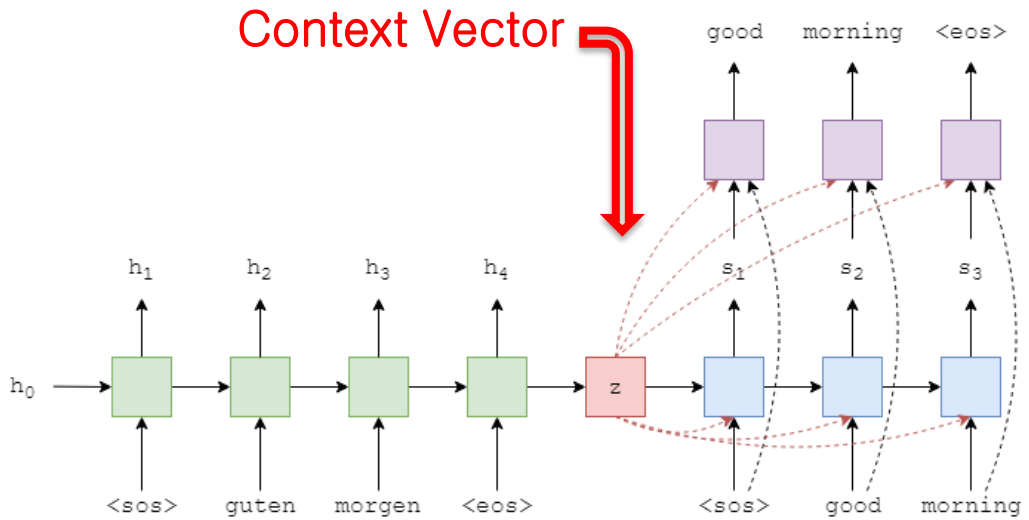
GRU

- Architecture

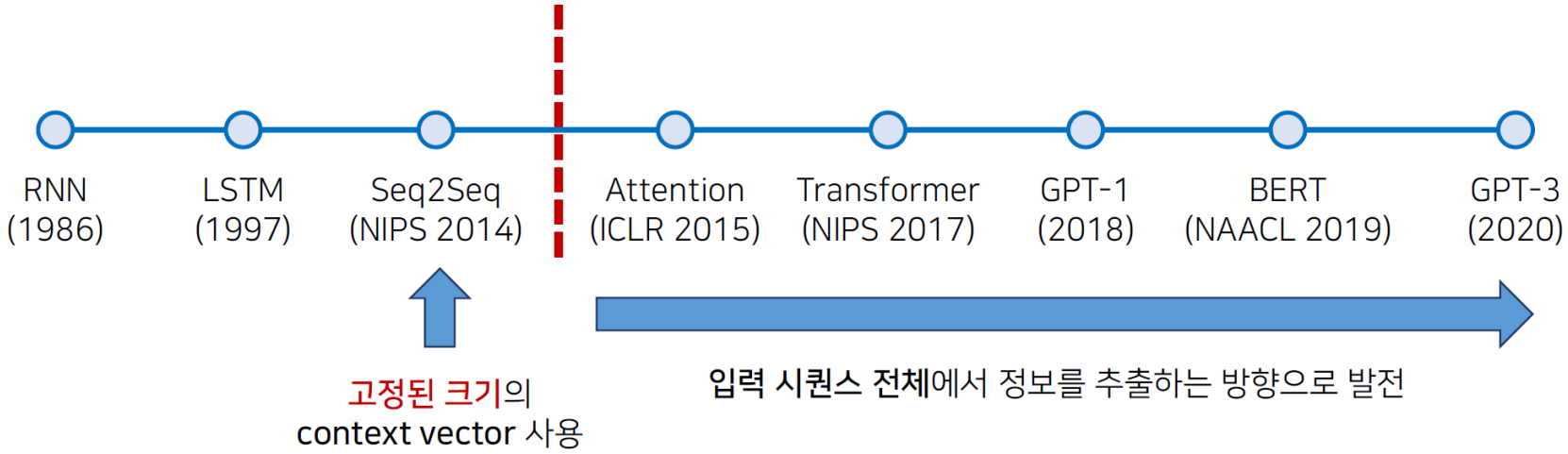


Seq2Seq


Context Vector
Er liebte zu essen .
Softmax
Encoder
S
Decoder
Embed
NULL Er liebte zu essen
He loved to eat .


Context Vector
good morning <eos>
h1 h2 h3 h4
s1 s2 s3
h0
z
<sos> guten morgen <eos>
<sos> good morning

Seq2Seq



- 출처 : 동빈나

Transformer

- Attention Is All You Need

Attention Is All You Need

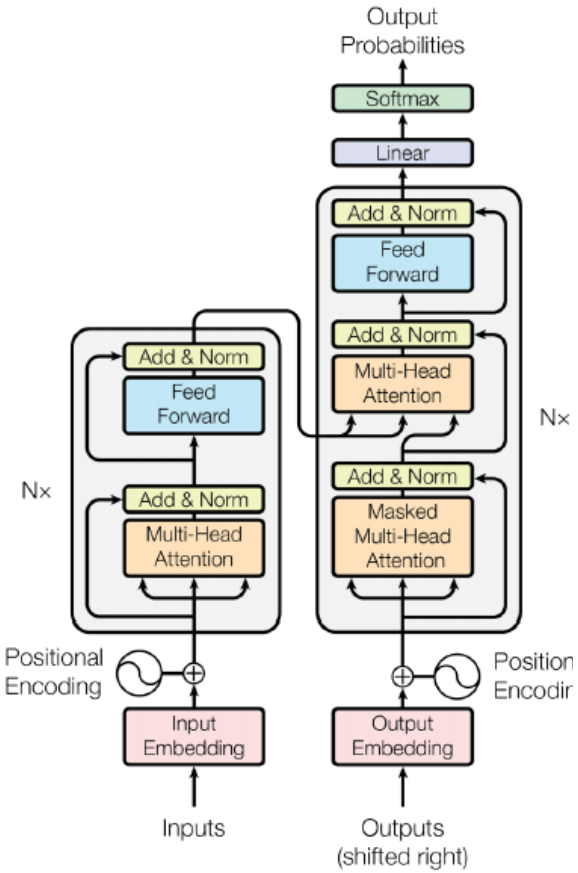
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

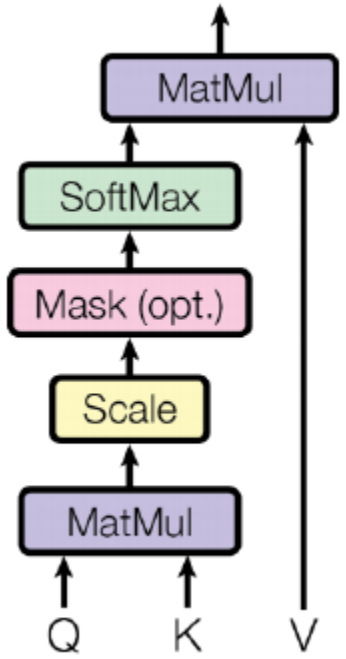
Abstract

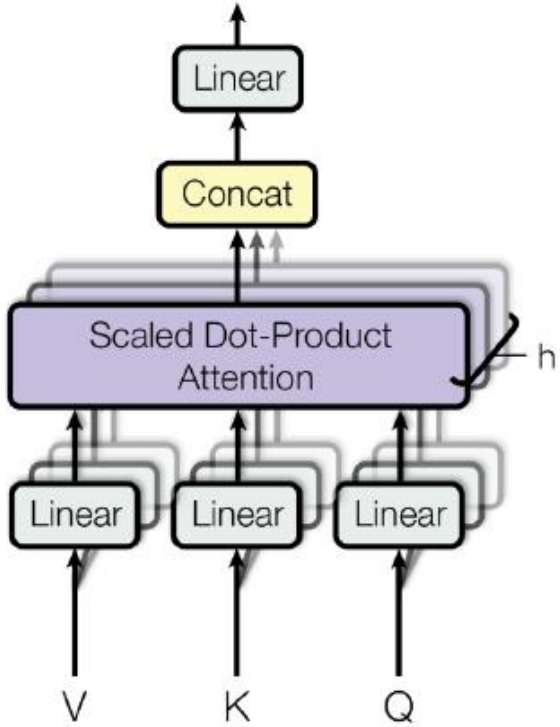
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer,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

Transformer

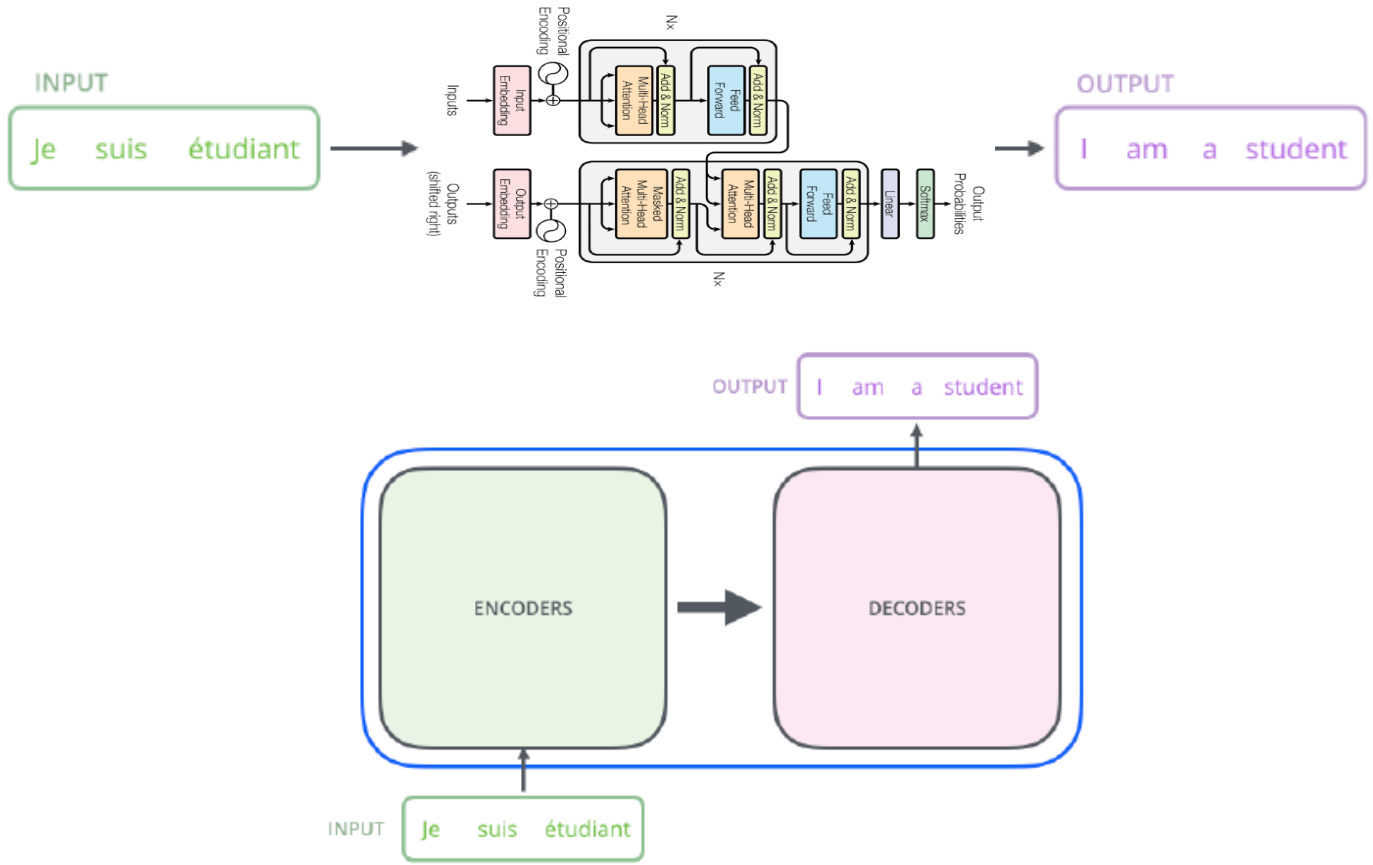
- Architecture





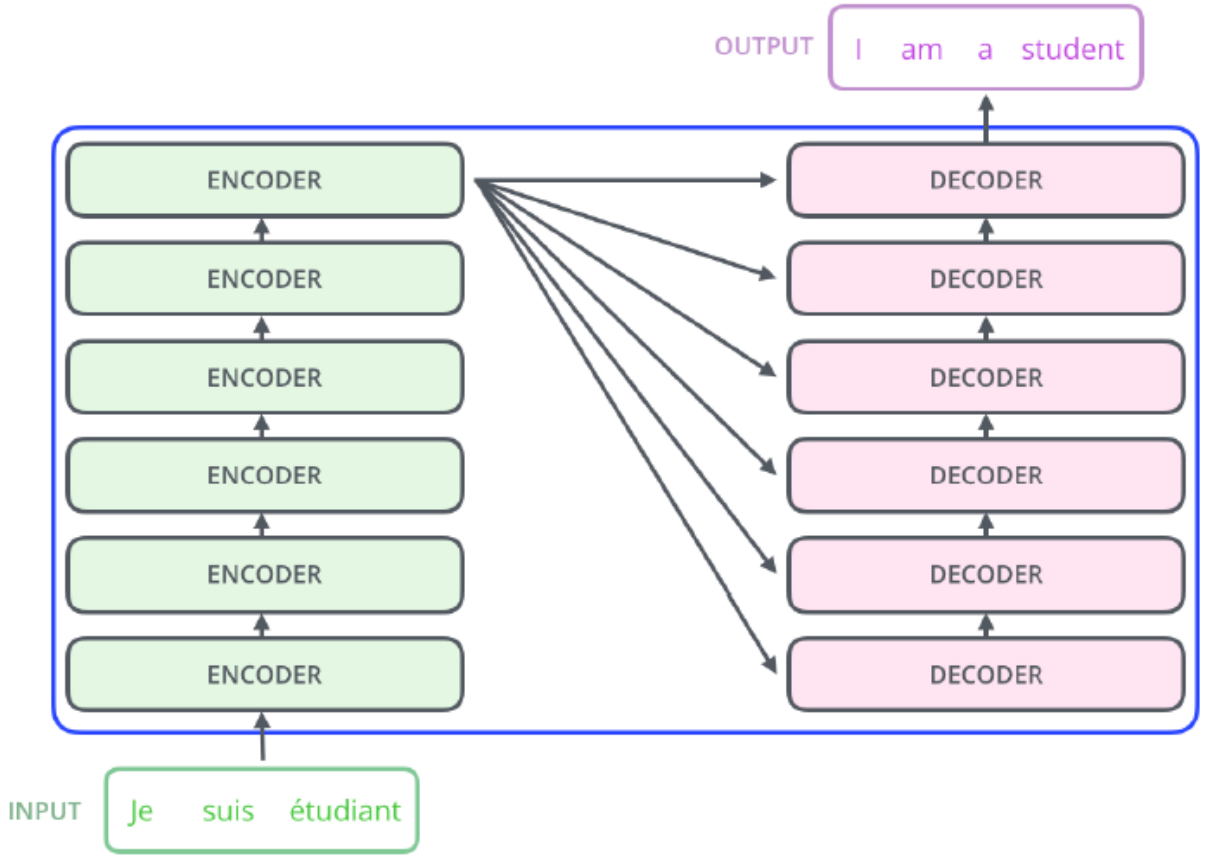


Transformer



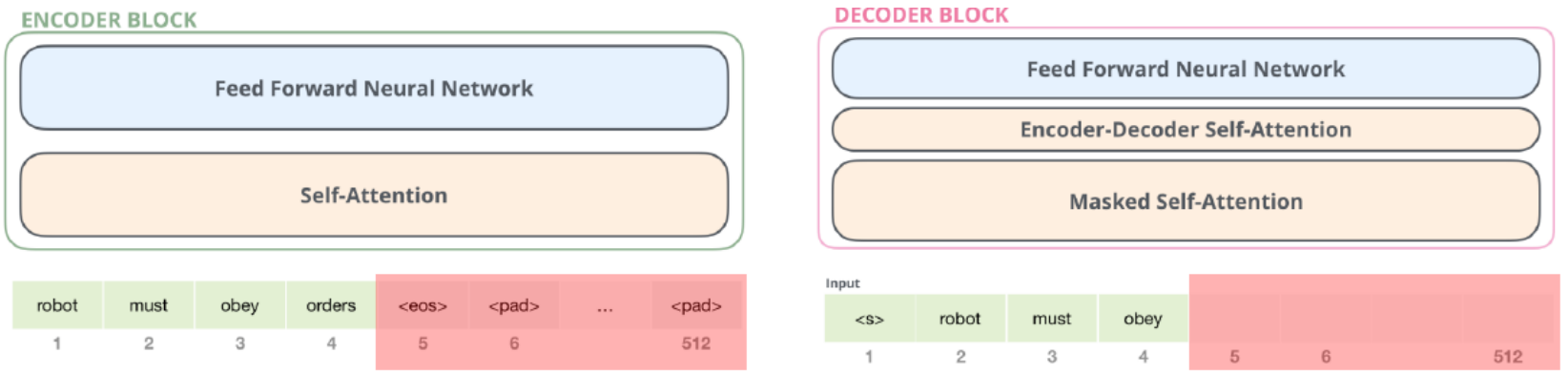
Transformer

- Architecture



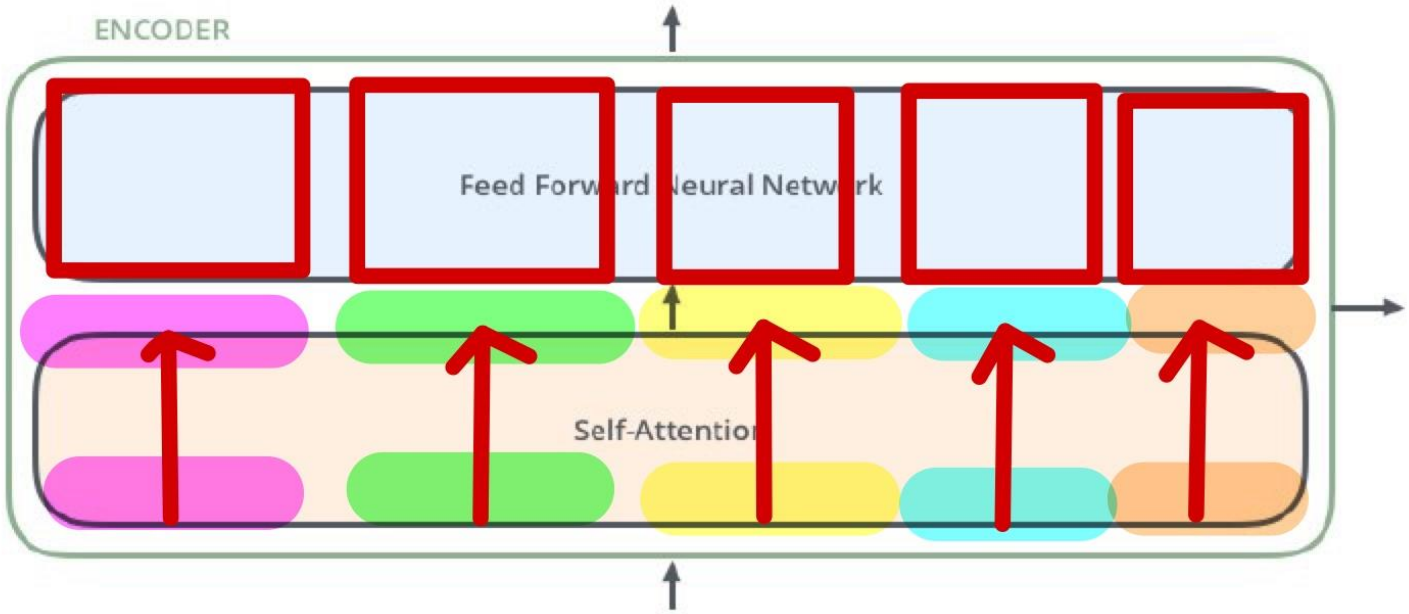
Transformer

- Architecture

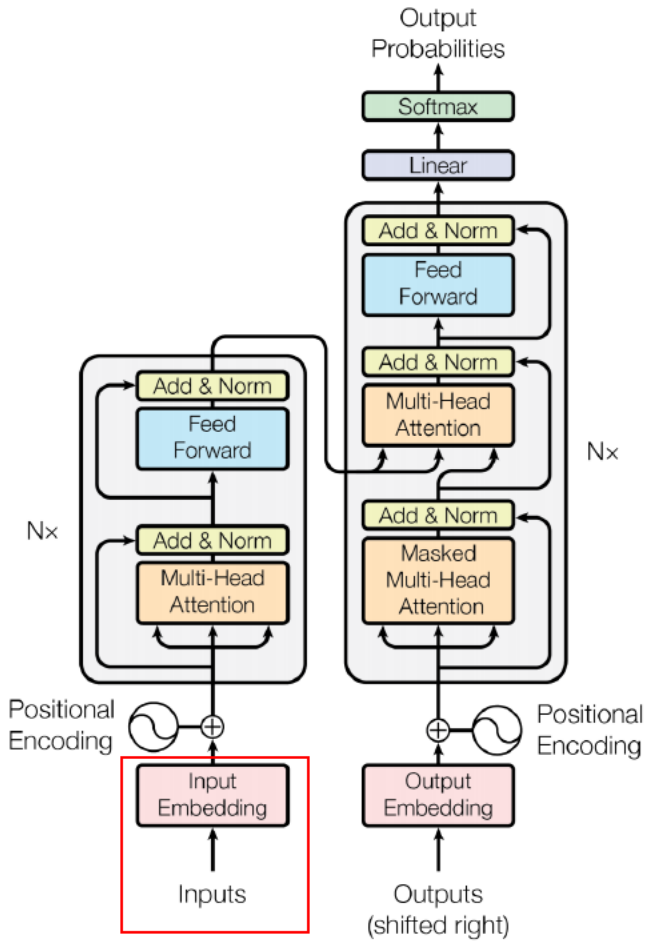


Transformer

- Encoder



Transformer

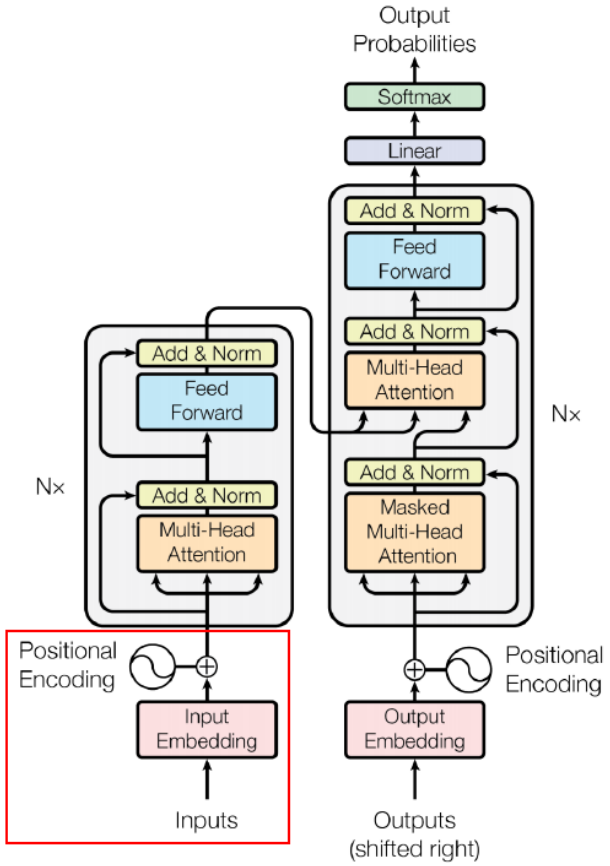


Transformer

- Word Embedding

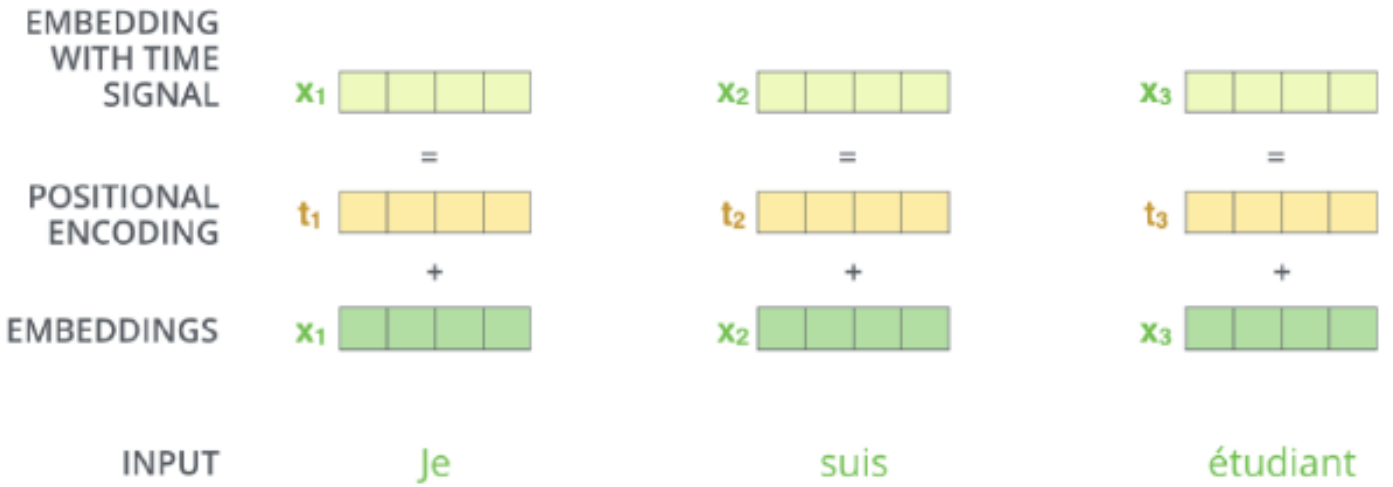
Transformer

- Positional Encoding



Transformer

- Positional Encoding



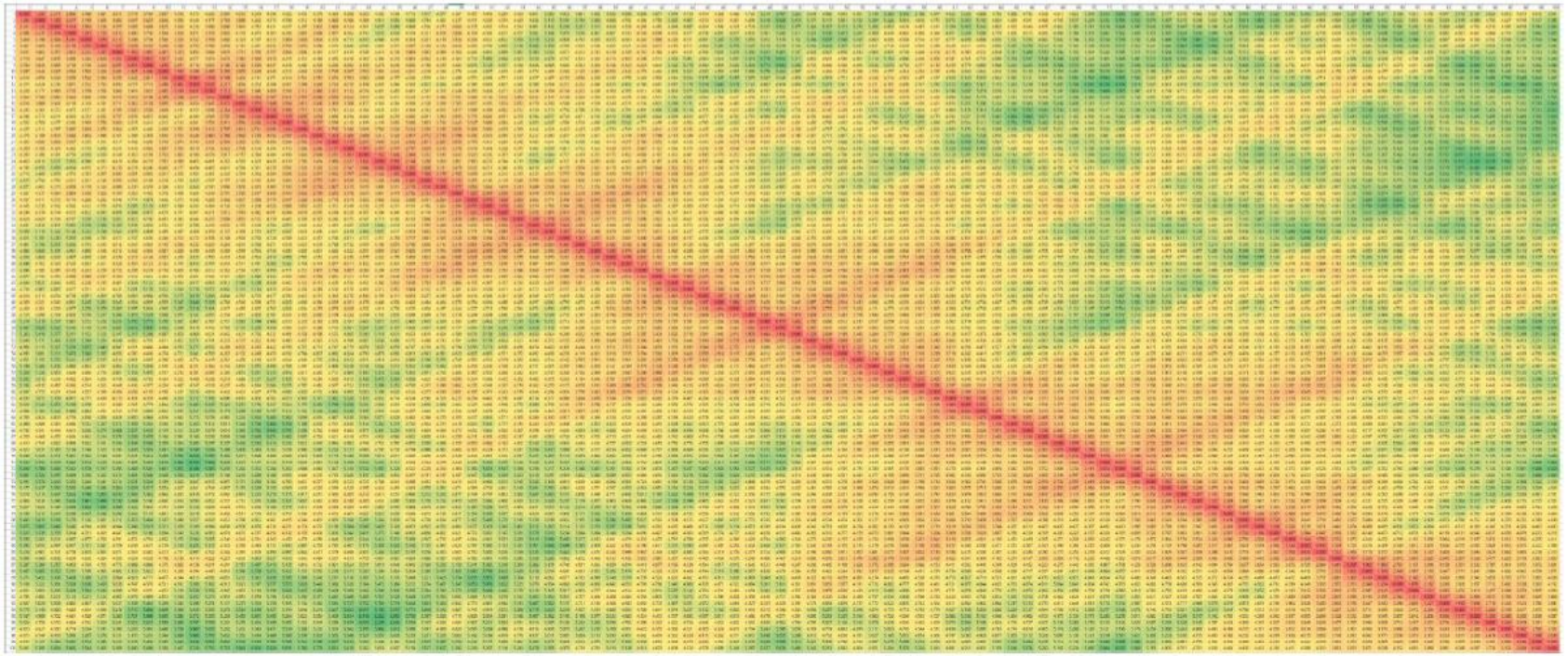
Transformer

- Positional Encoding

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X1	0.000	1.275	2.167	2.823	3.361	3.508	3.392	3.440	3.417	3.266
X2	1.275	0.000	1.104	2.195	3.135	3.511	3.452	3.442	3.387	3.308
X3	2.167	1.104	0.000	1.296	2.468	3.067	3.256	3.464	3.498	3.371
X4	2.823	2.195	1.296	0.000	1.275	2.110	2.746	3.399	3.624	3.399
X5	3.361	3.135	2.468	1.275	0.000	1.057	2.176	3.242	3.659	3.434
X6	3.508	3.511	3.067	2.110	1.057	0.000	1.333	2.601	3.169	3.118
X7	3.392	3.452	3.256	2.746	2.176	1.333	0.000	1.338	2.063	2.429
X8	3.440	3.442	3.464	3.399	3.242	2.601	1.338	0.000	0.912	1.891
X9	3.417	3.387	3.498	3.624	3.659	3.169	2.063	0.912	0.000	1.277
X10	3.266	3.308	3.371	3.399	3.434	3.118	2.429	1.891	1.277	0.000

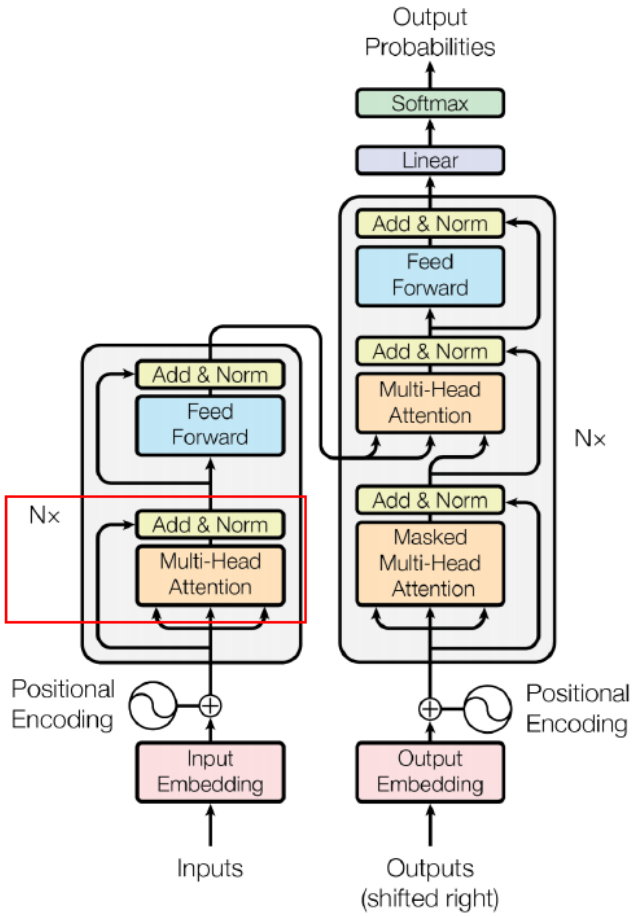
Transformer

- Positional Encoding



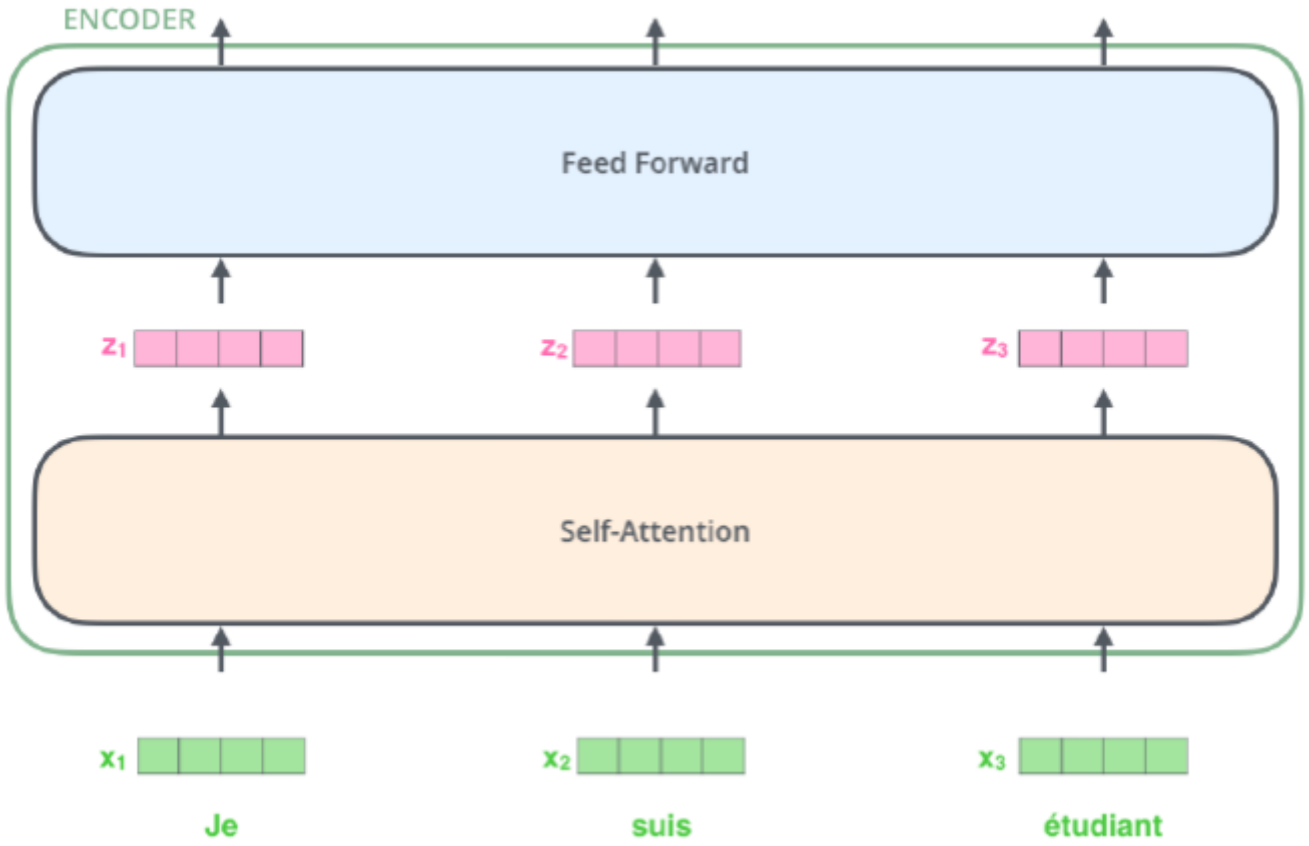
Transformer

- Multi-Head Attention



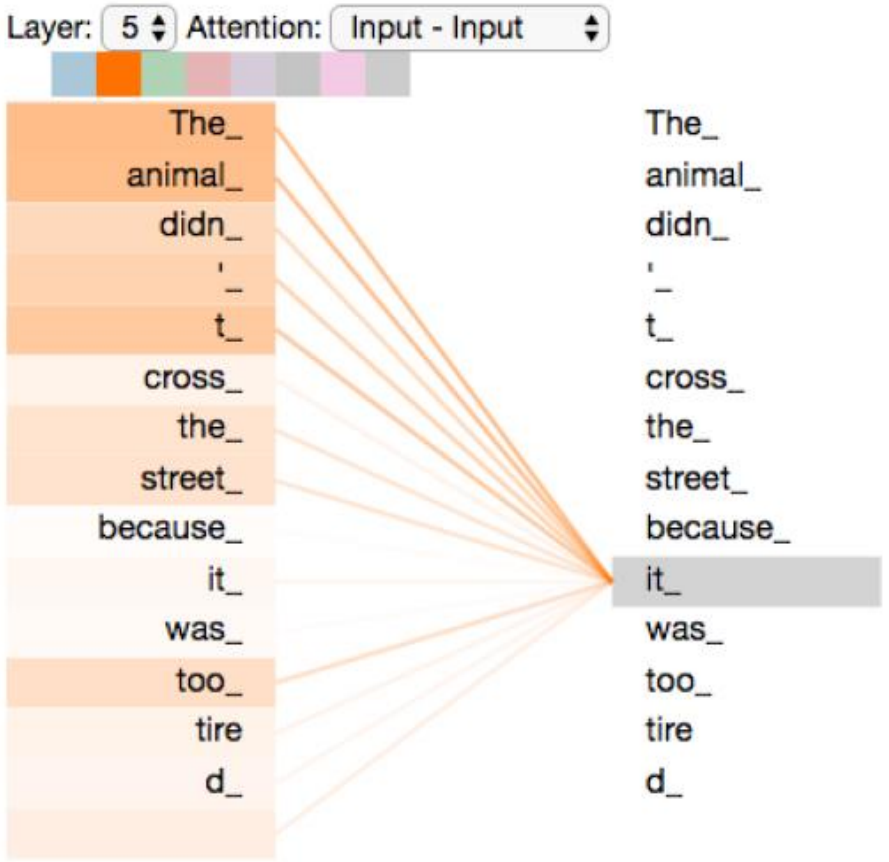
Transformer

- Encoder



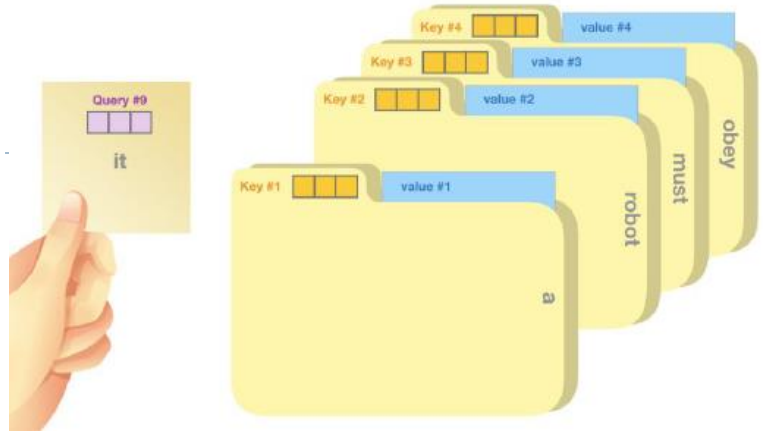
Transformer

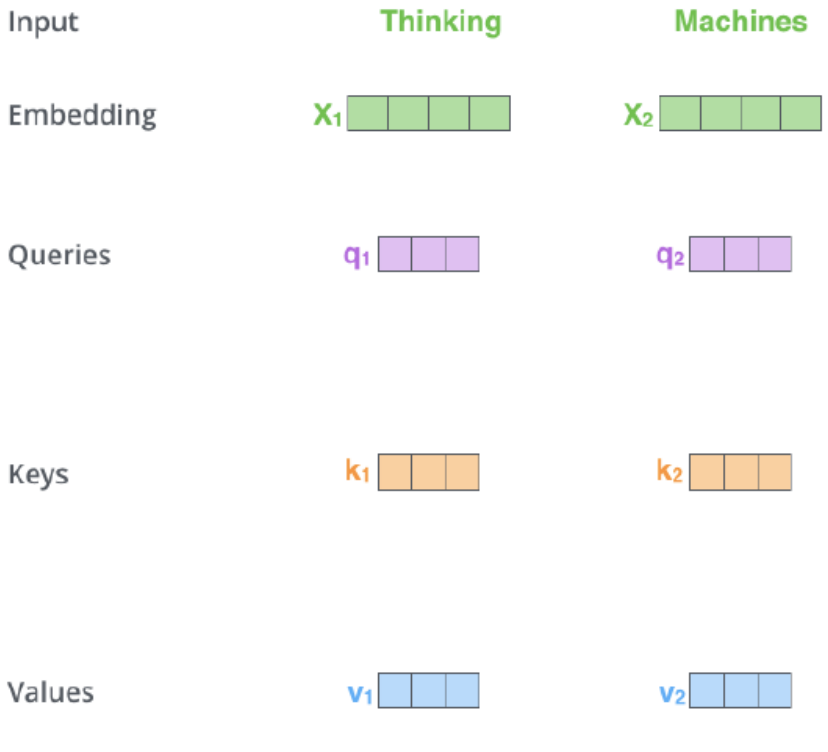
- Encoder

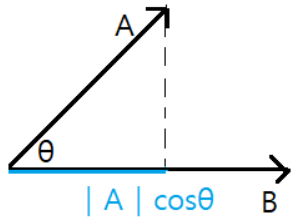


Transformer

- Attention

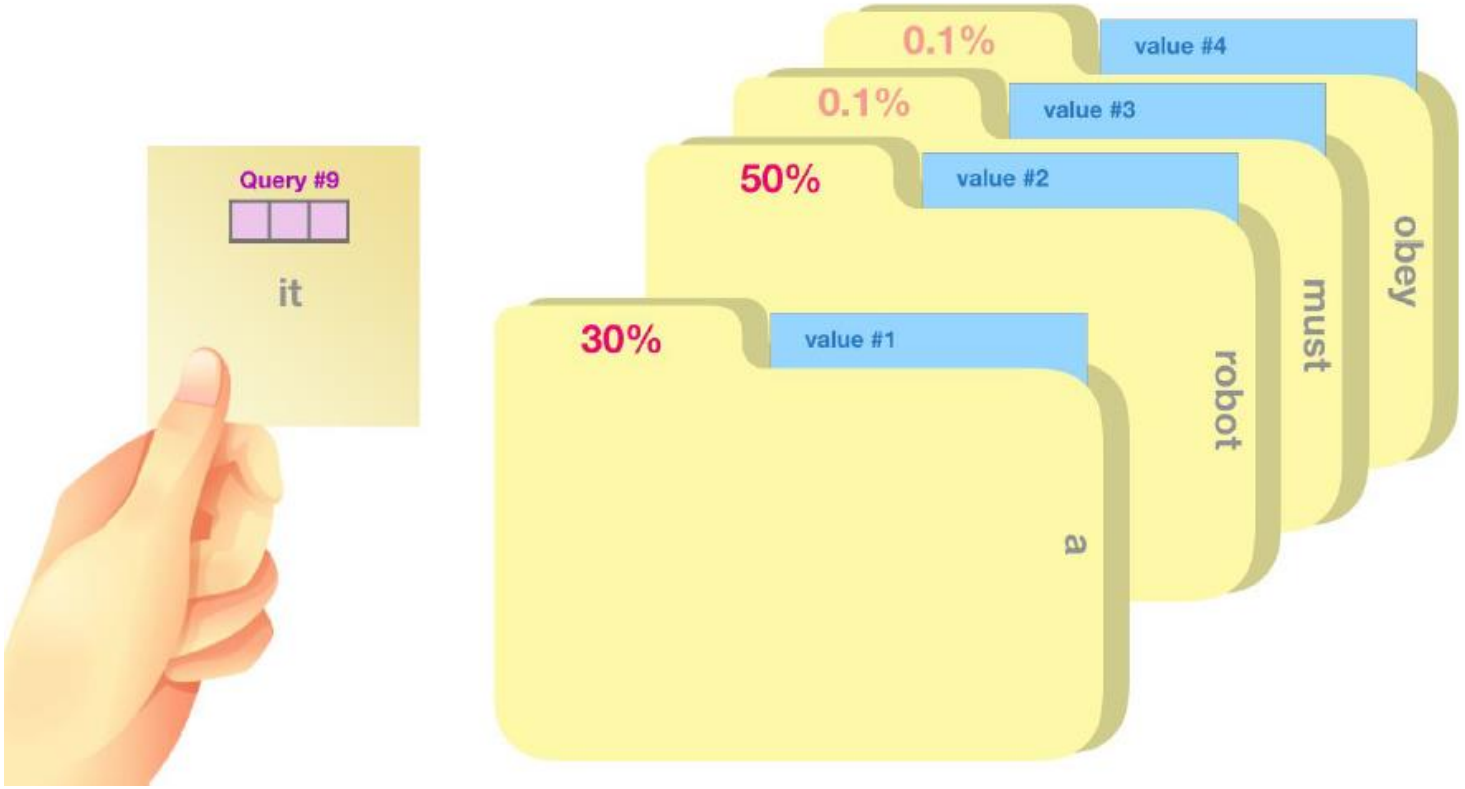






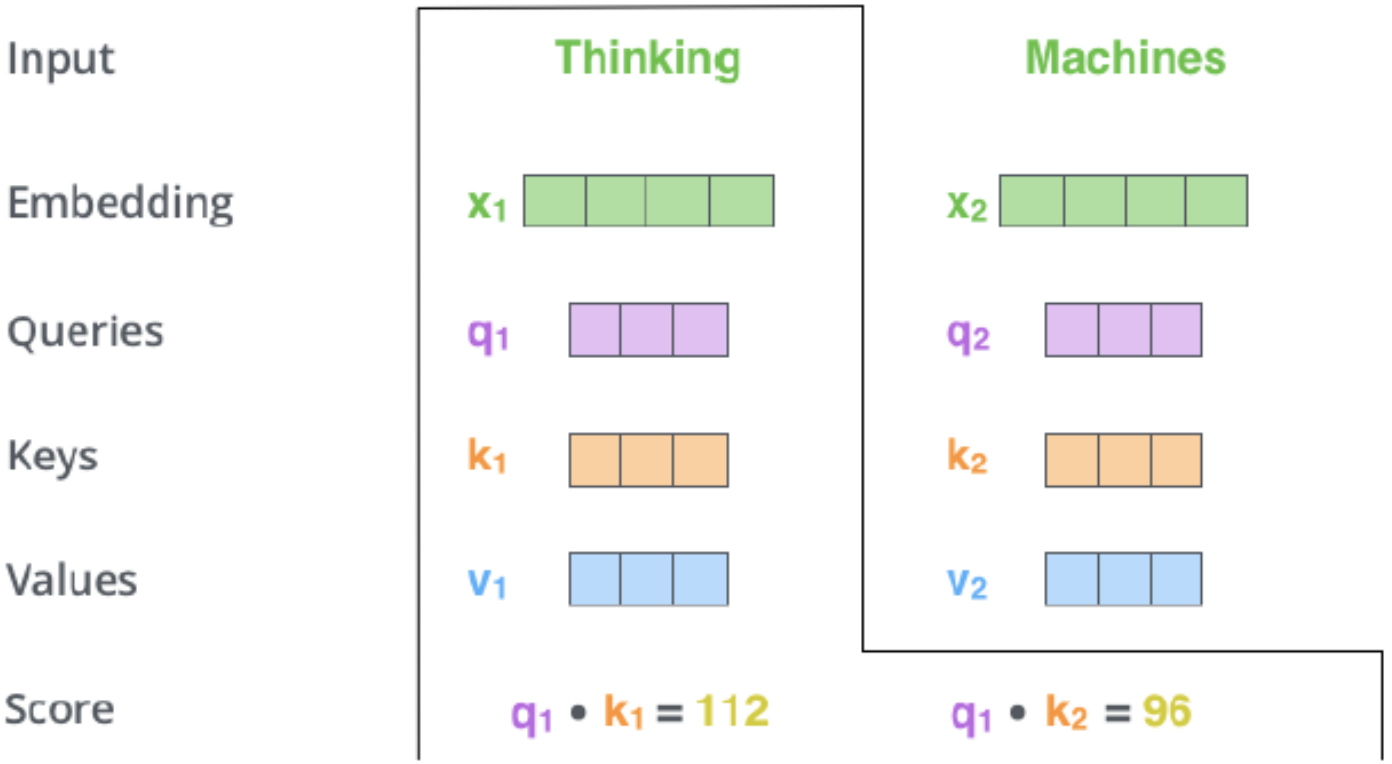
Transformer

- Attention



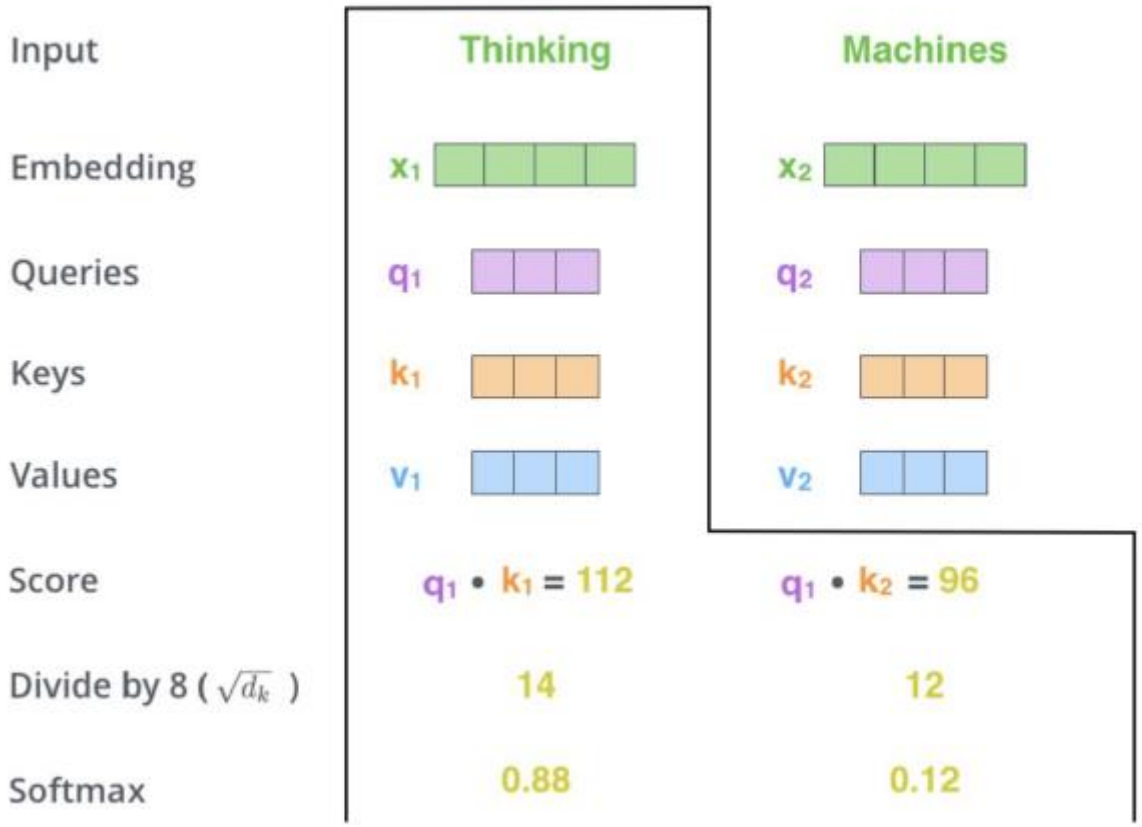
Transformer

- Attention



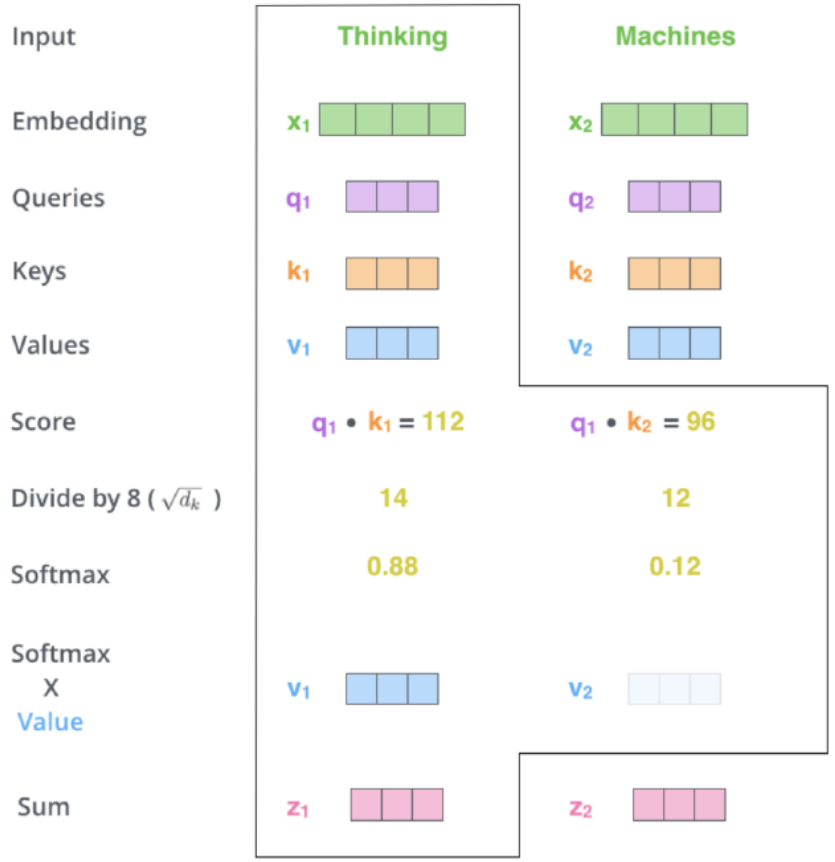
Transformer

- Attention



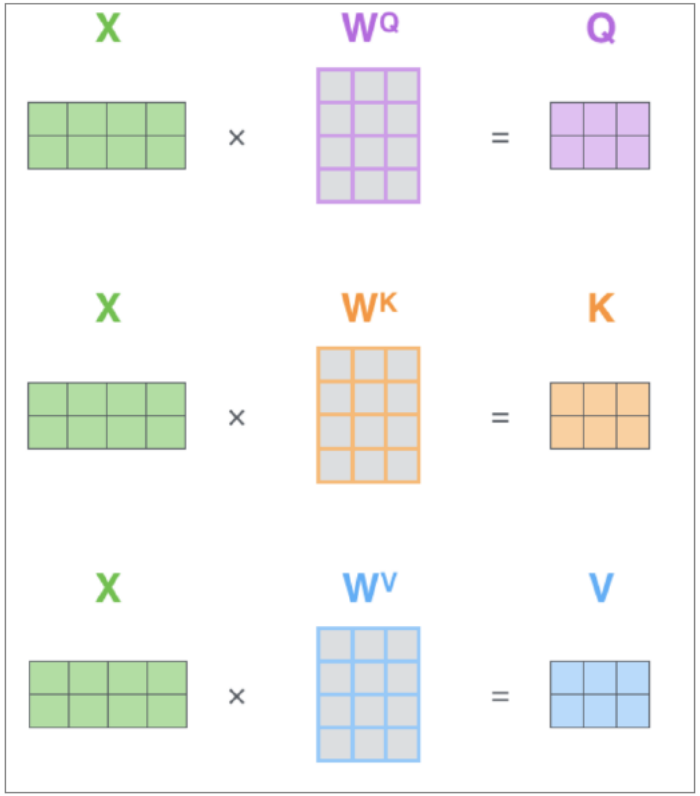
Transformer

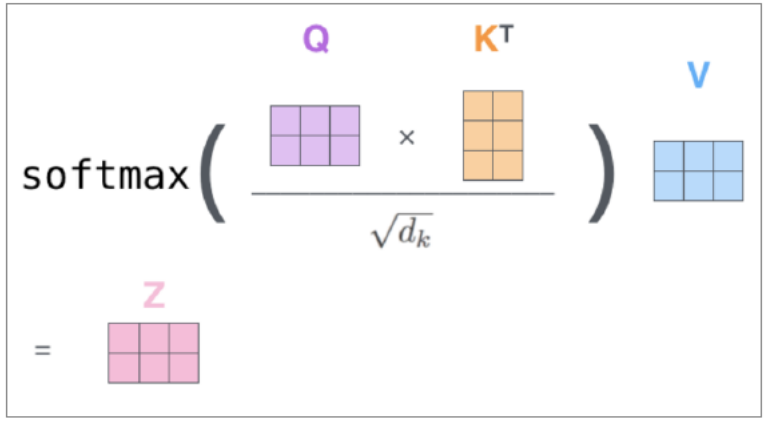
- Attention



Transformer

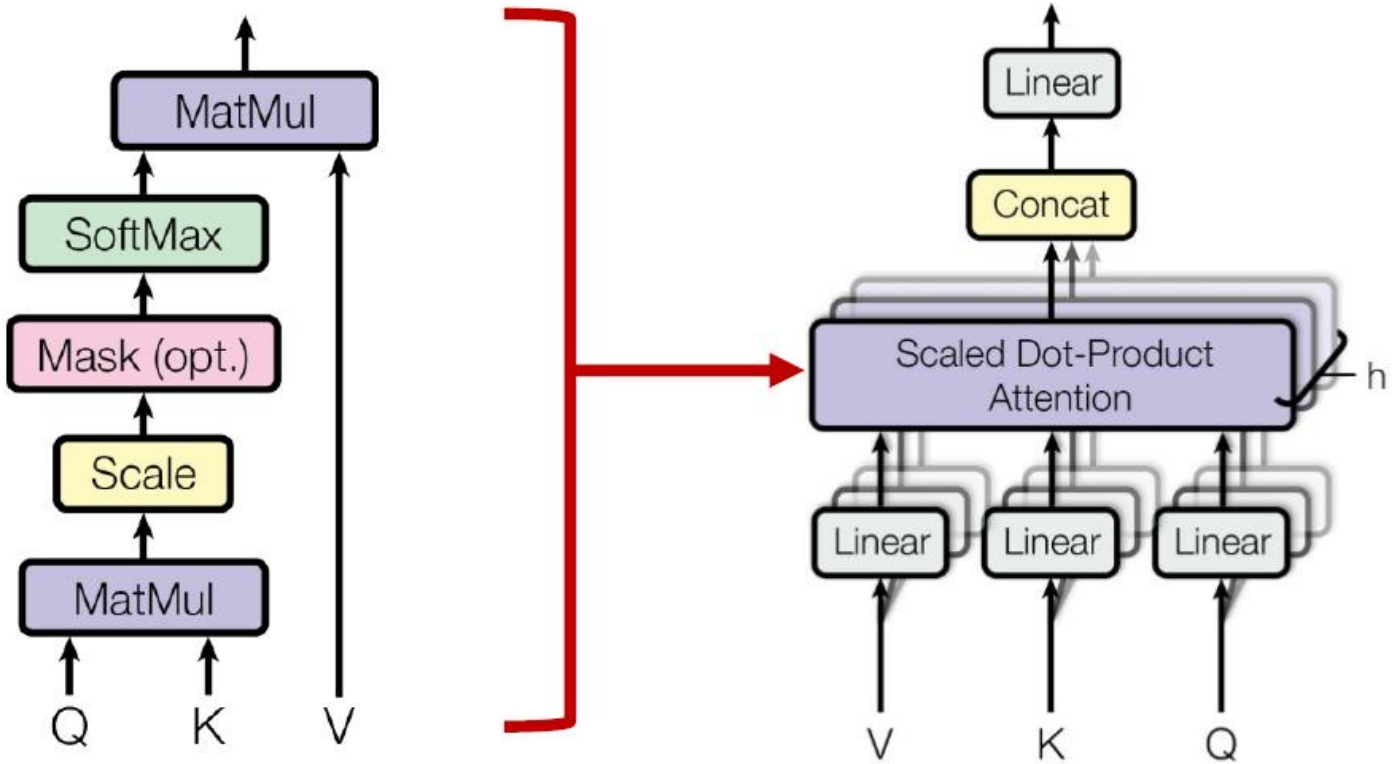
- Attention





Transformer

- Attention

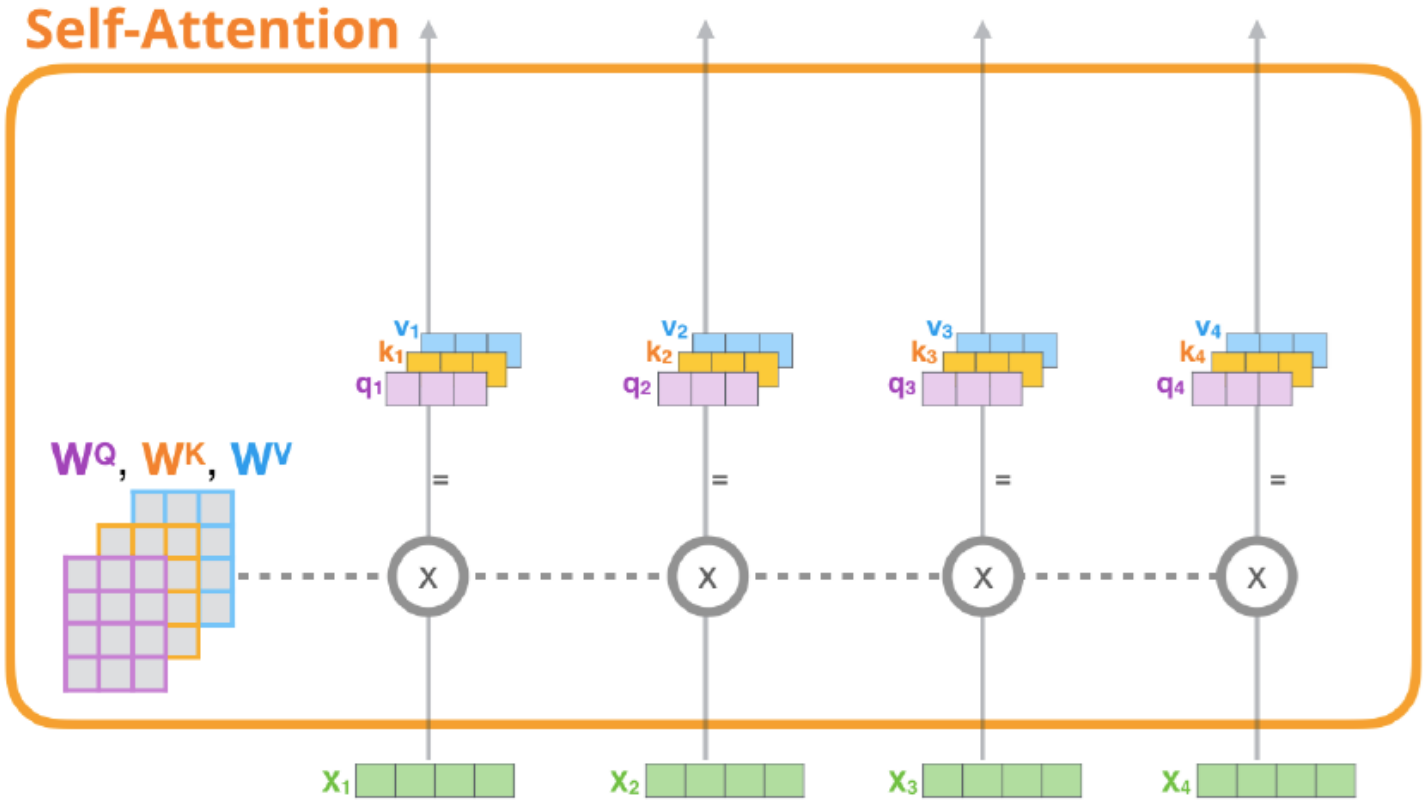


Scaled Dot-Product Attention

Multi-Head Attention

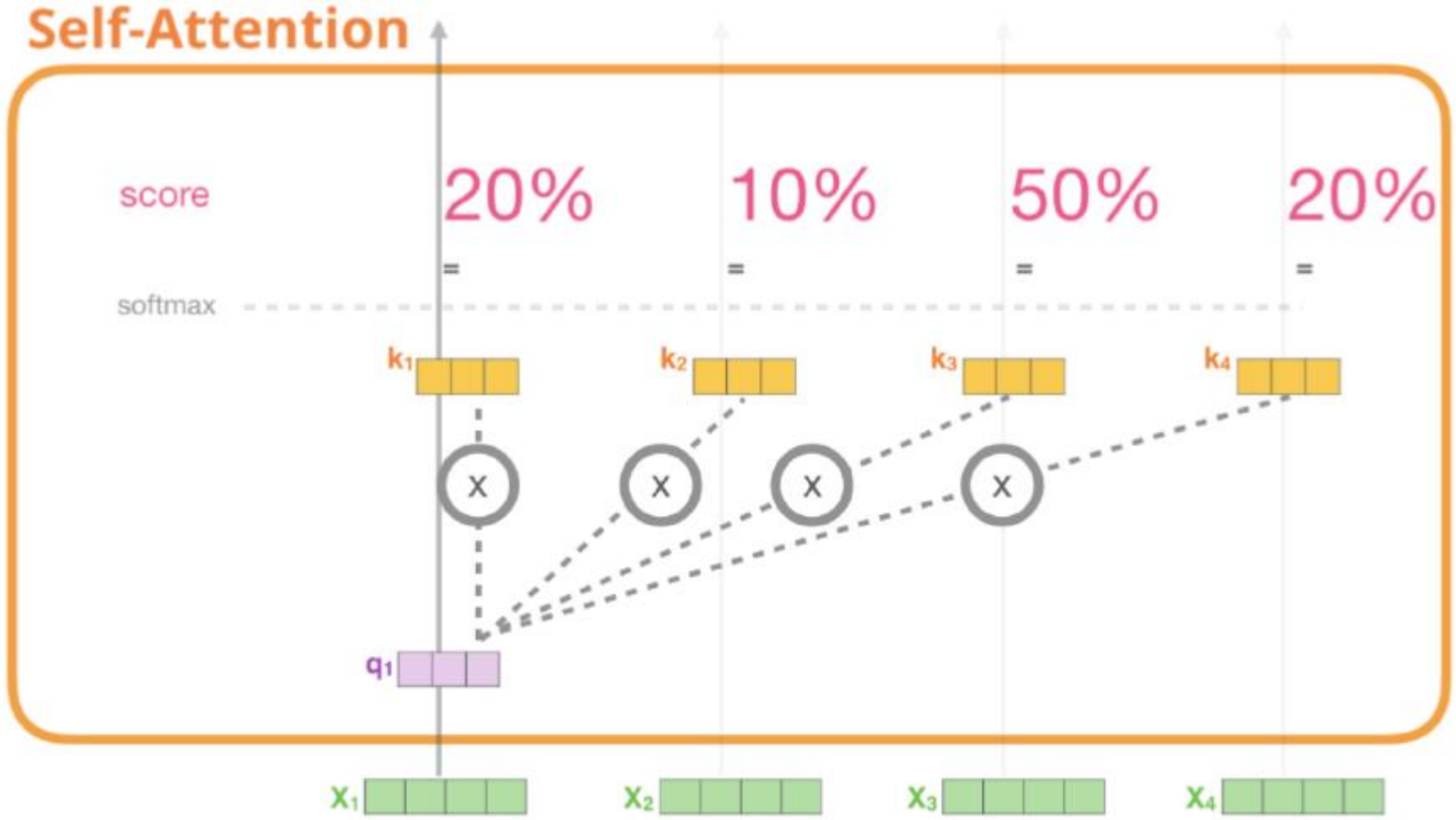
Transformer

- Attention



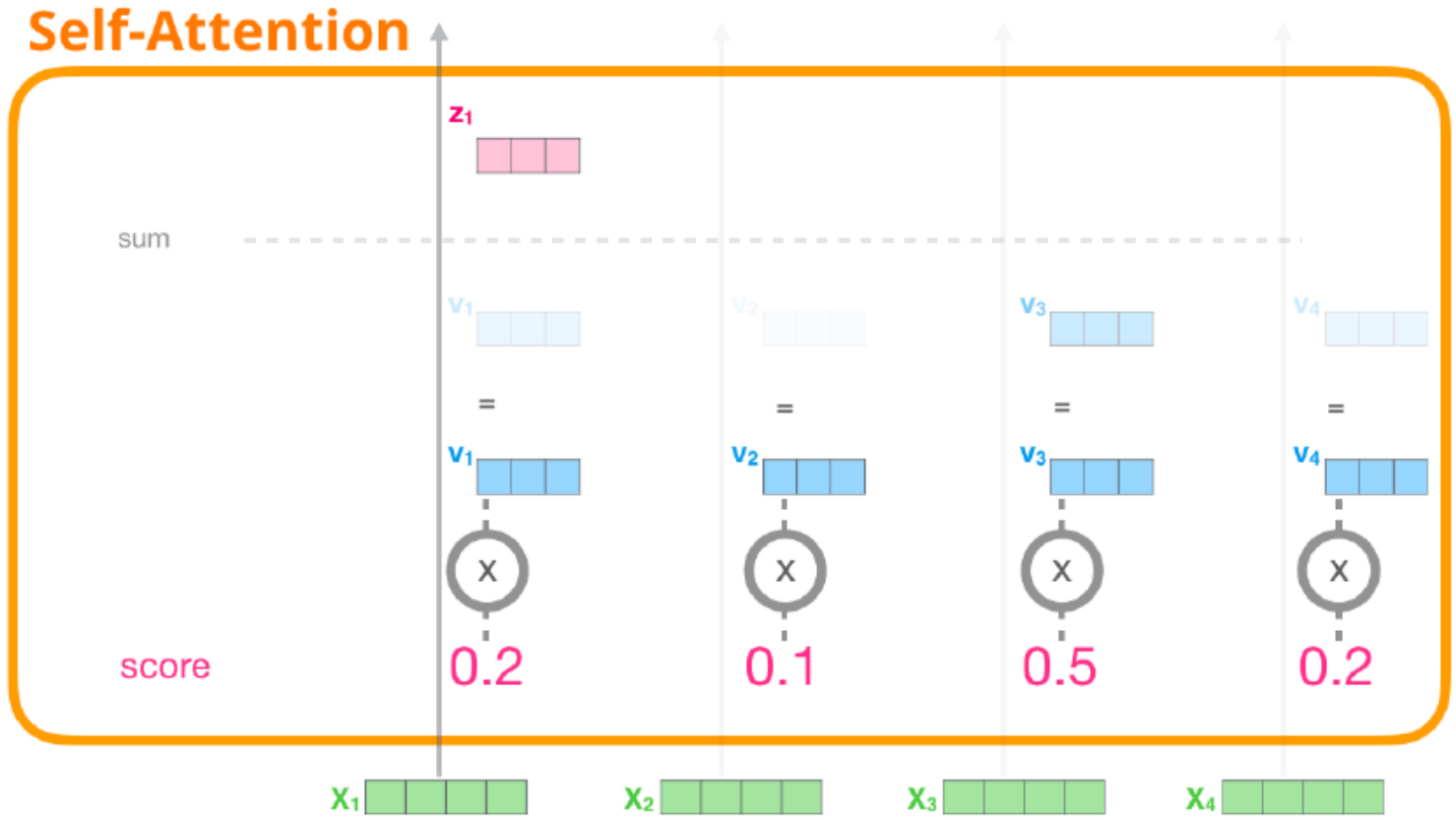
Transformer

- Attention



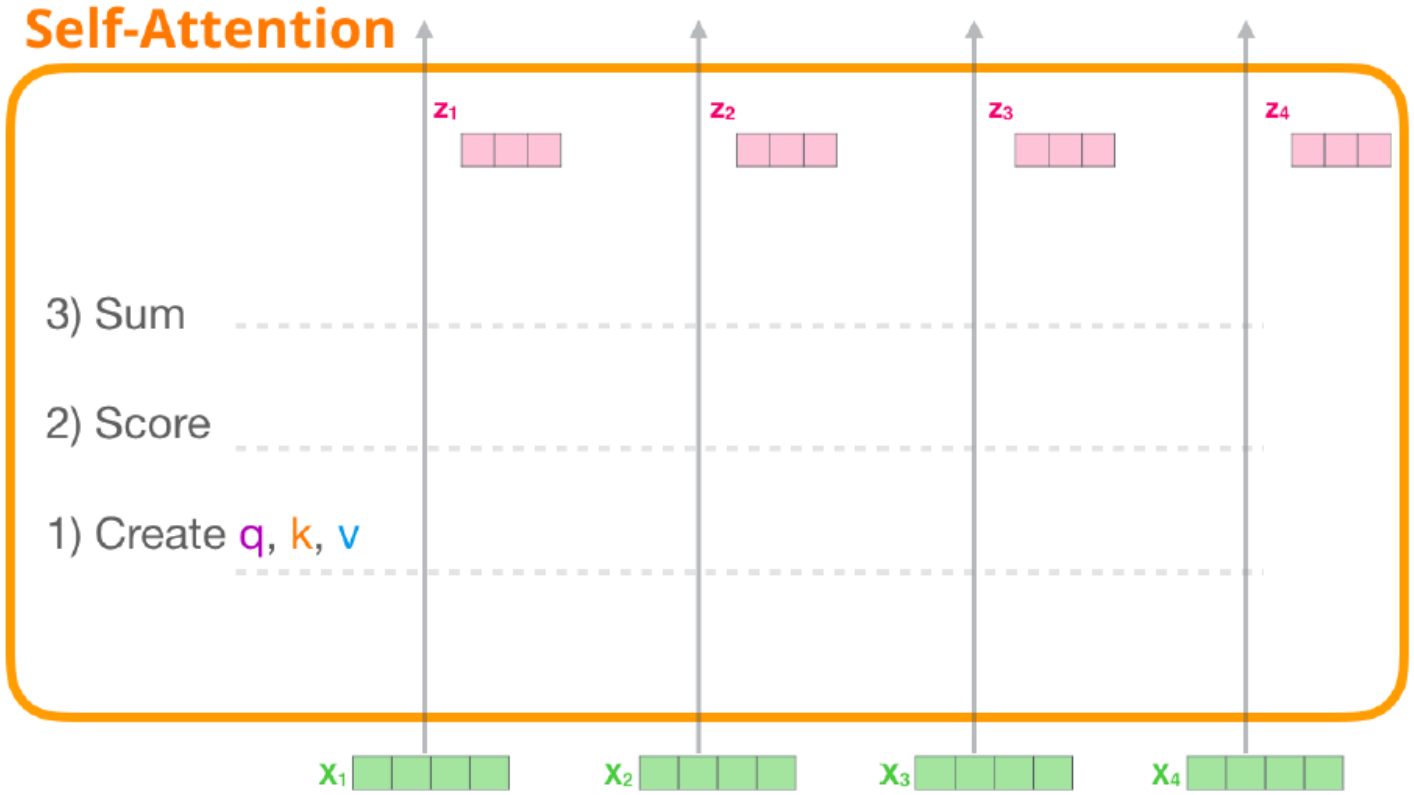
Transformer

- Attention



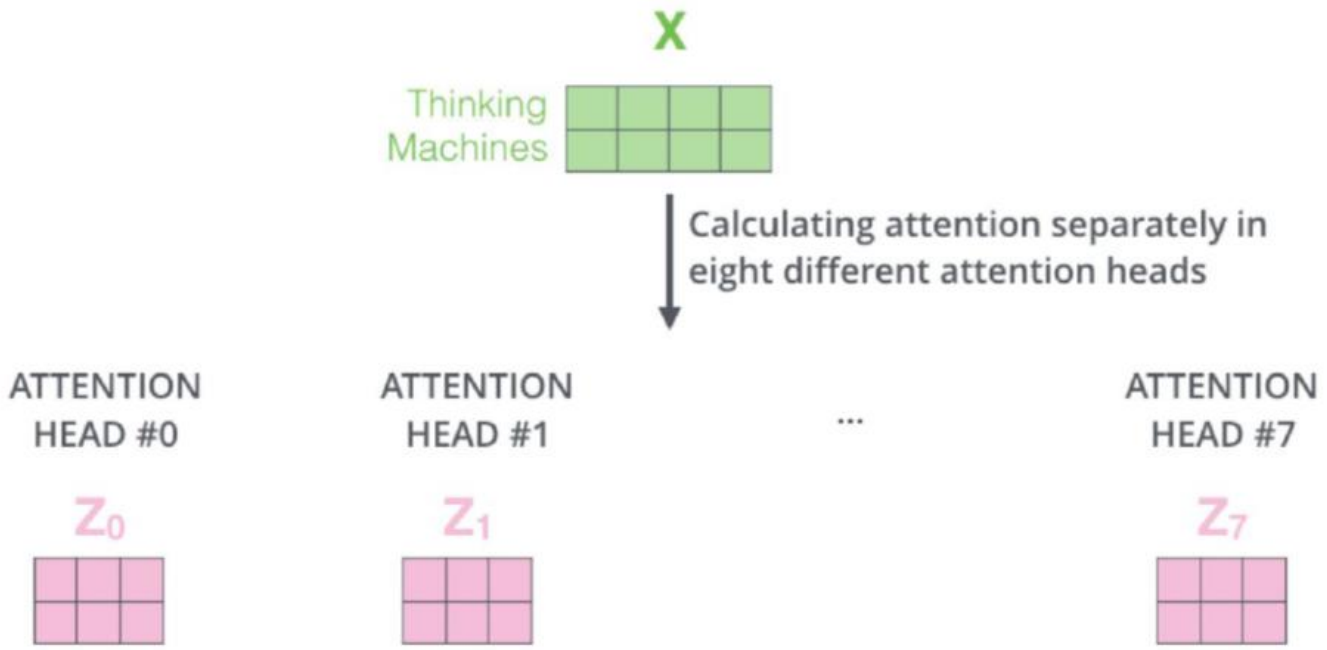
Transformer

- Attention



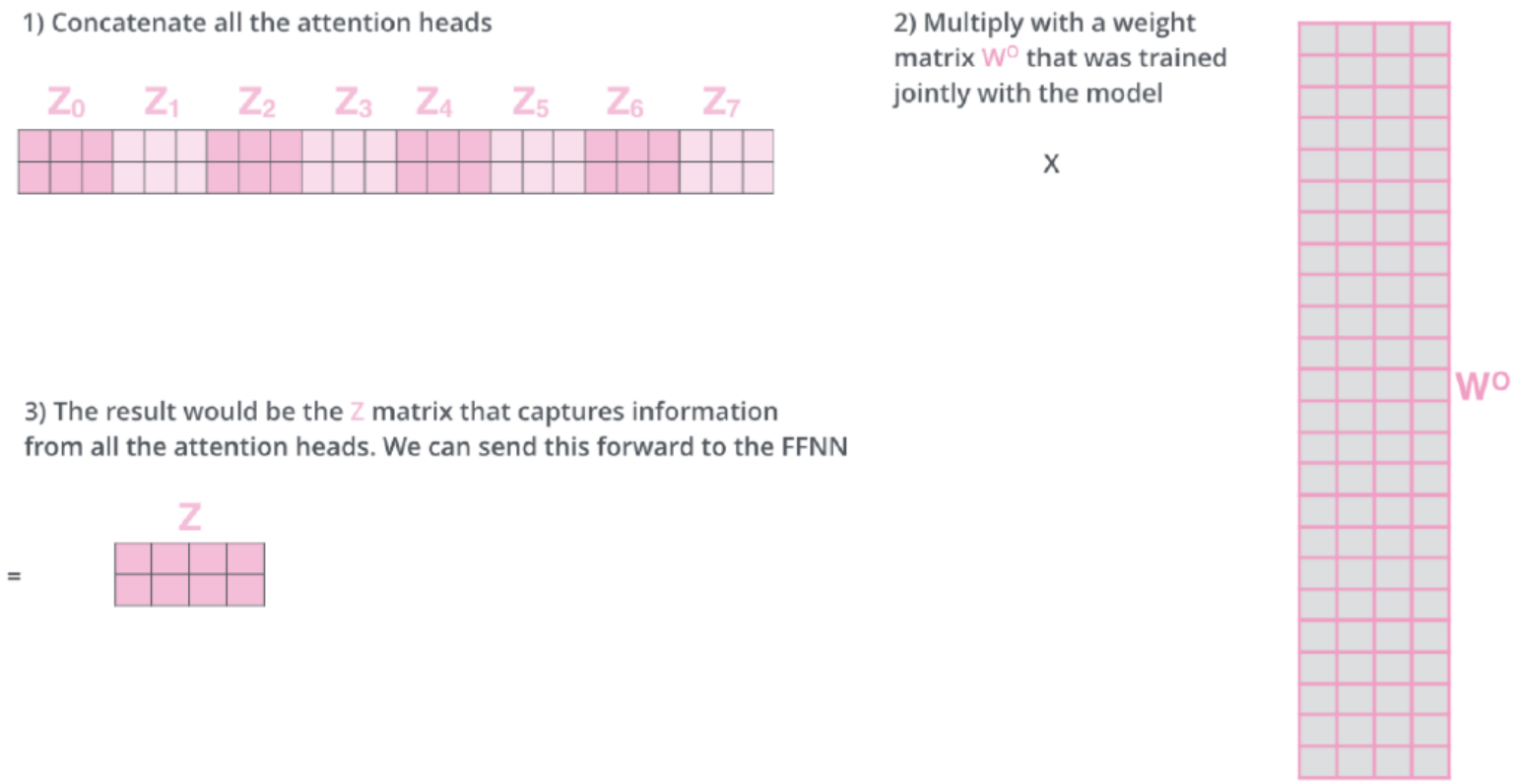
Transformer

- Attention



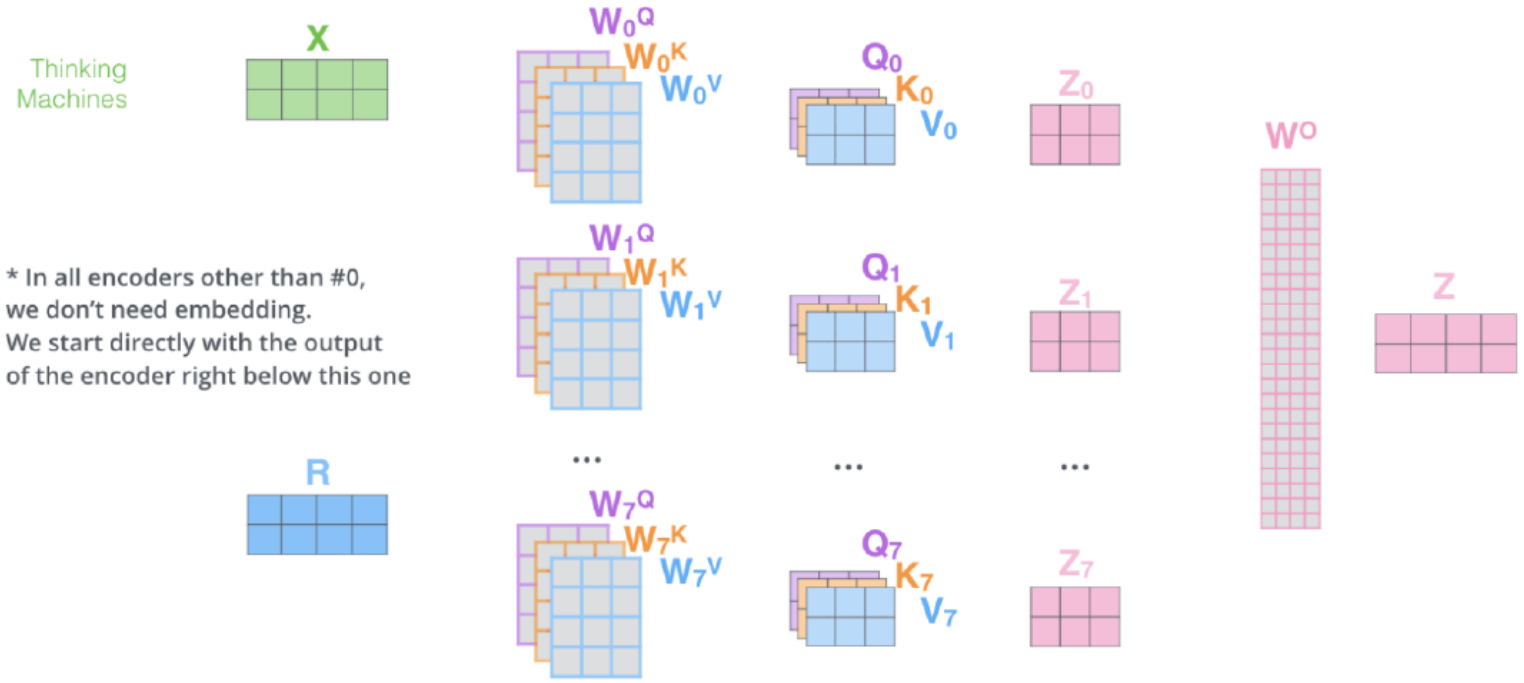
Transformer

- Attention



Transformer

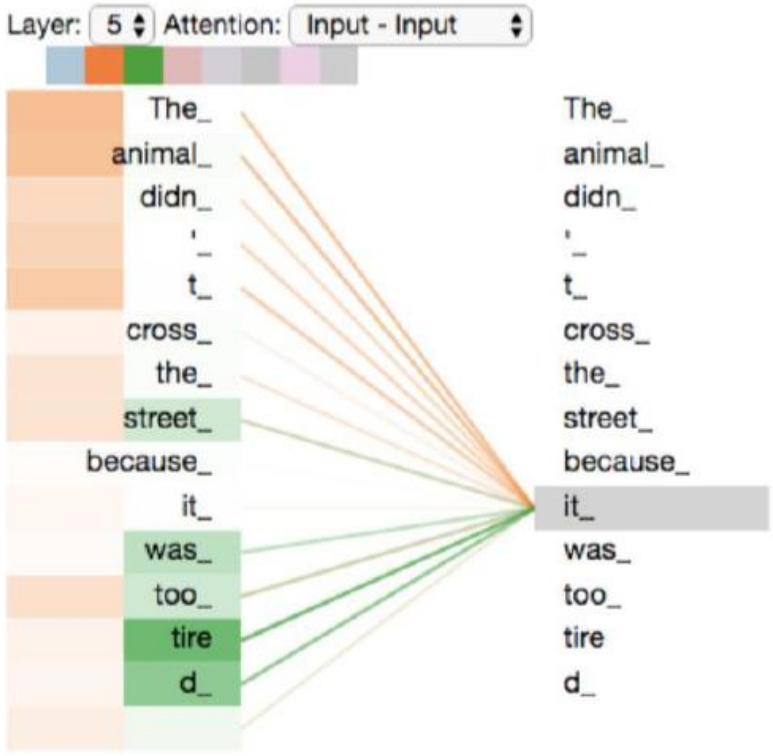
- Attention



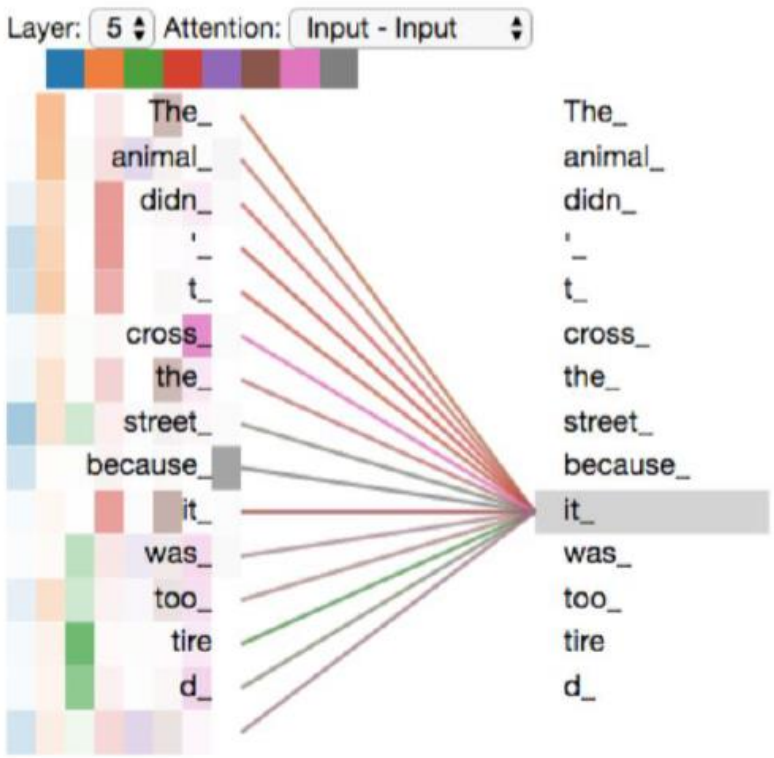
Transformer

- Multi Attention

Attention with two heads

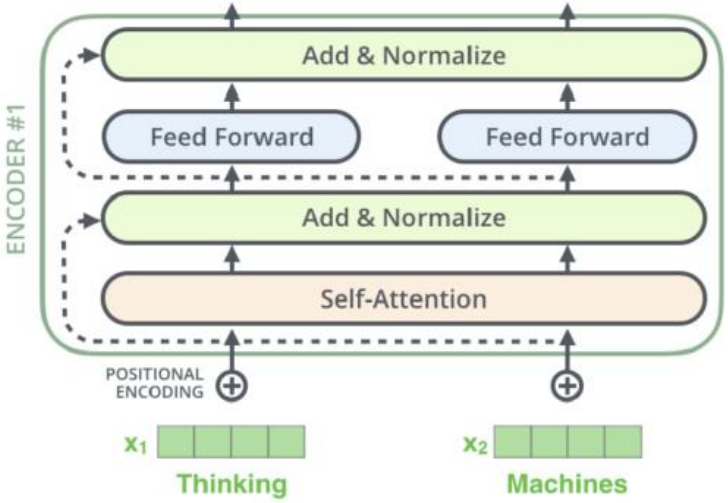


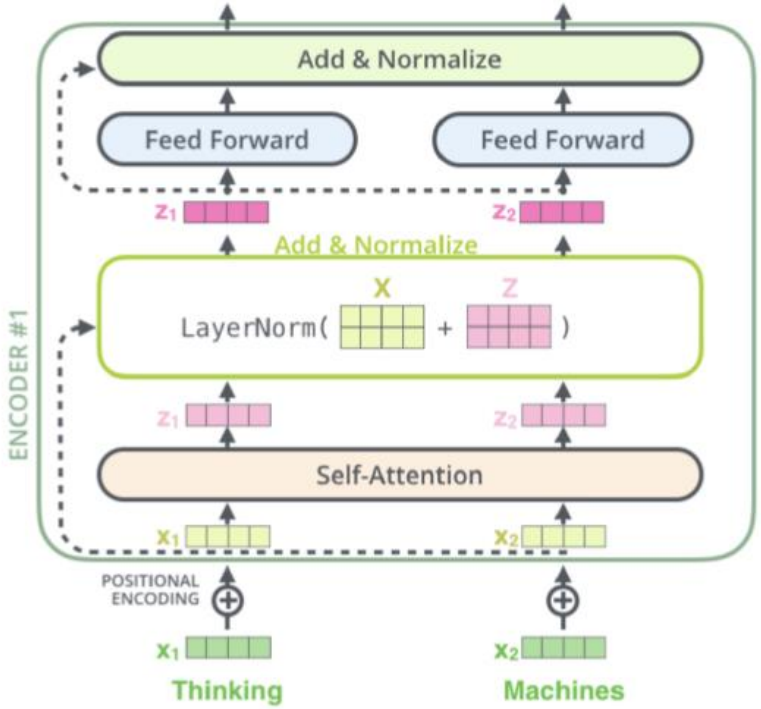
Attention with eight heads



Transformer

- Skip Connection and Normalization

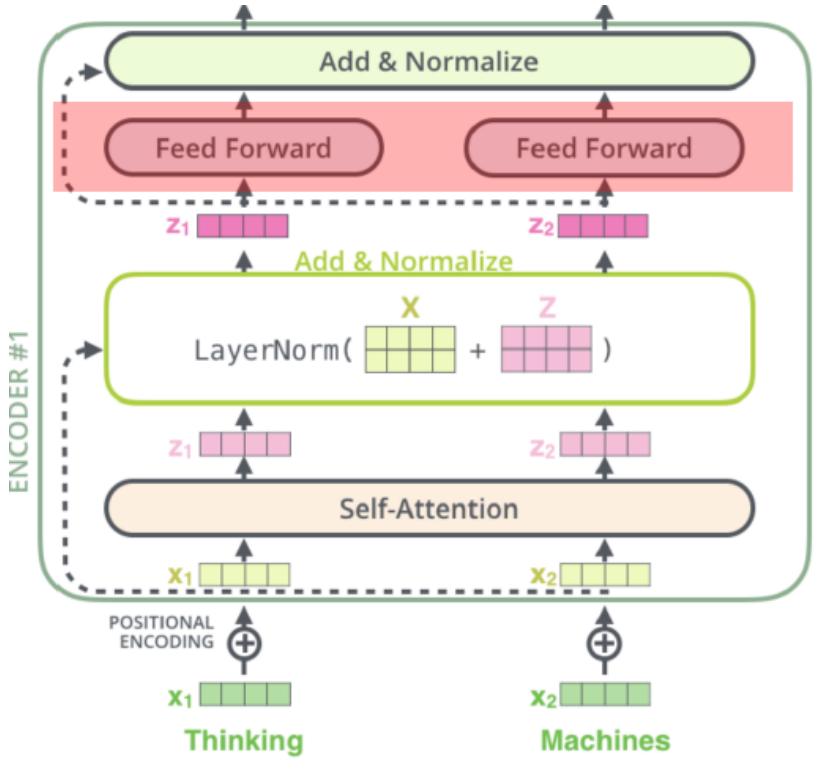




Transformer

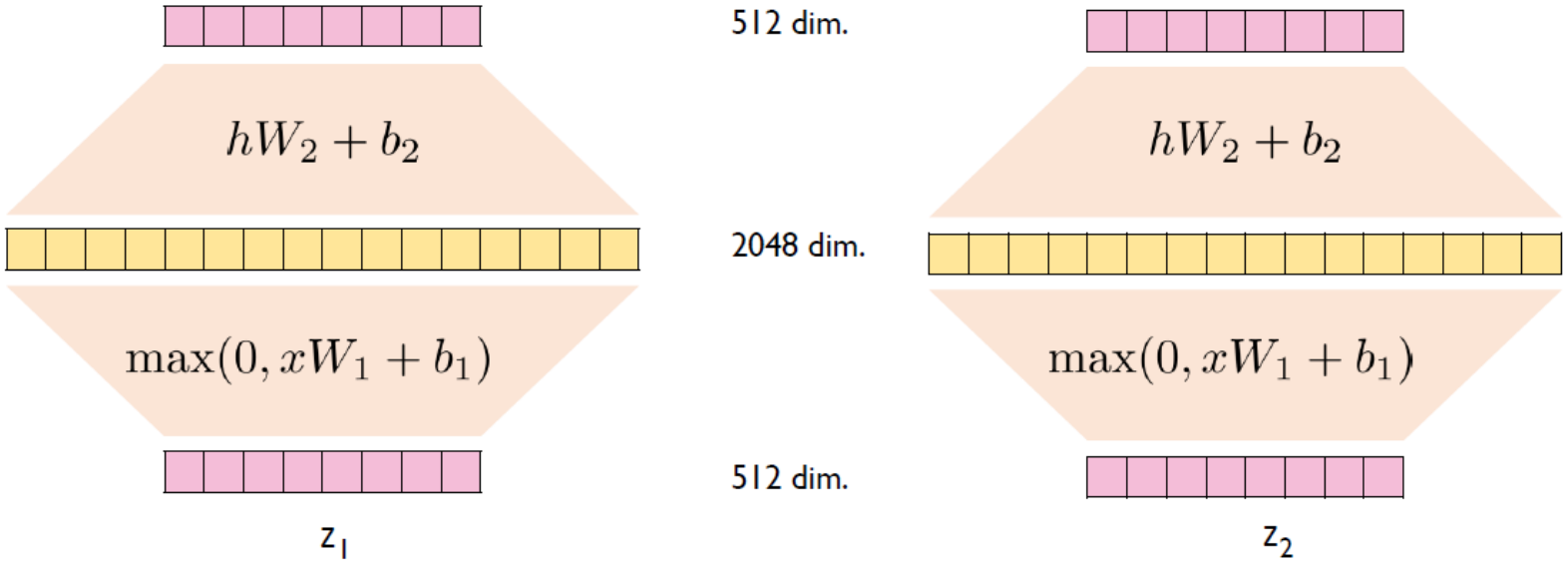
- Feed Forward Neural Network

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$



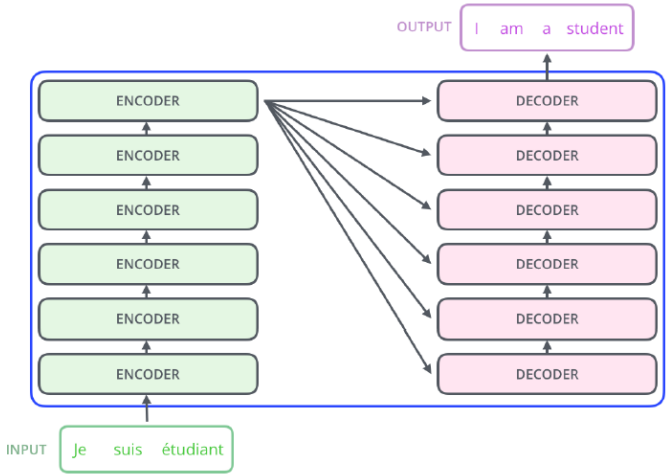
Transformer

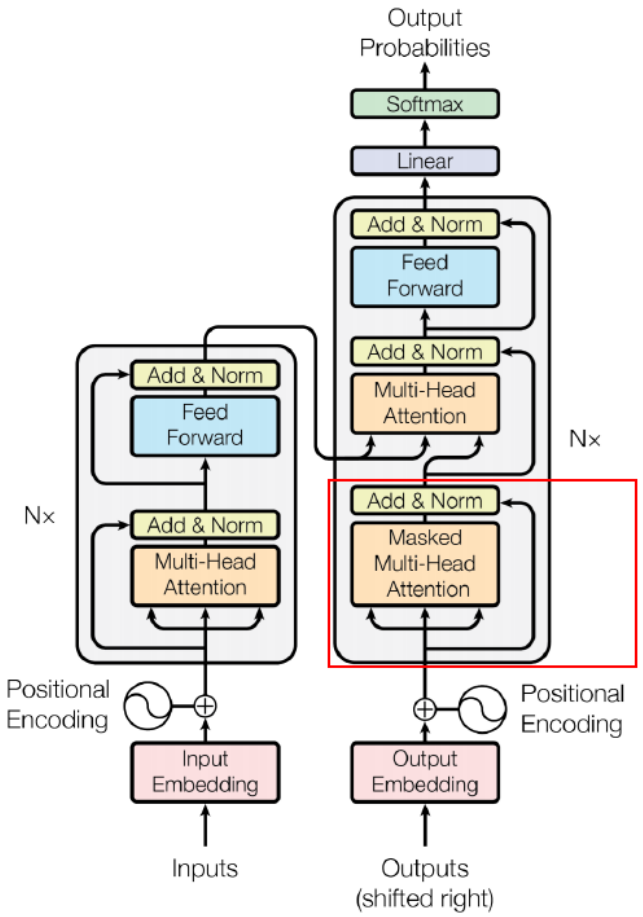
- Feed Forward Neural Network



Transformer

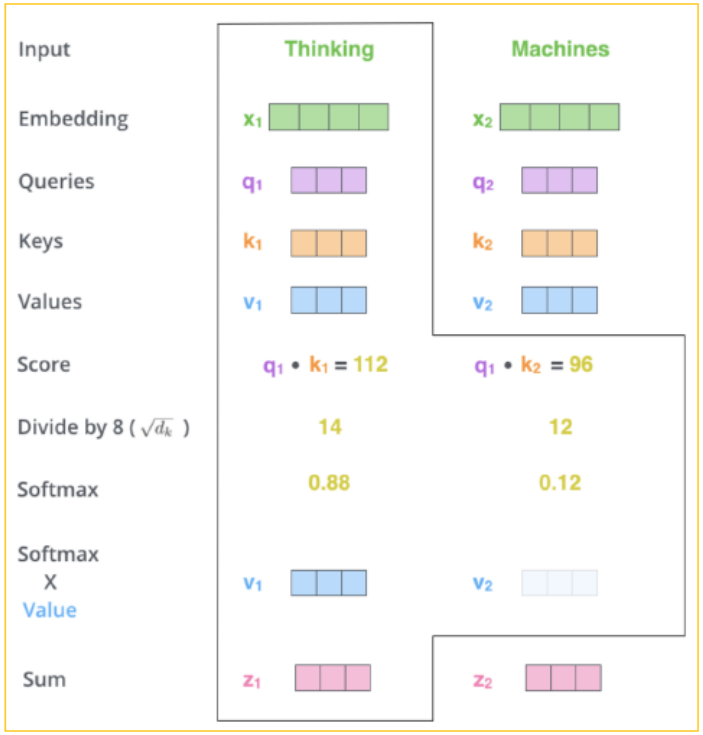
- Masked Multi Head Attention

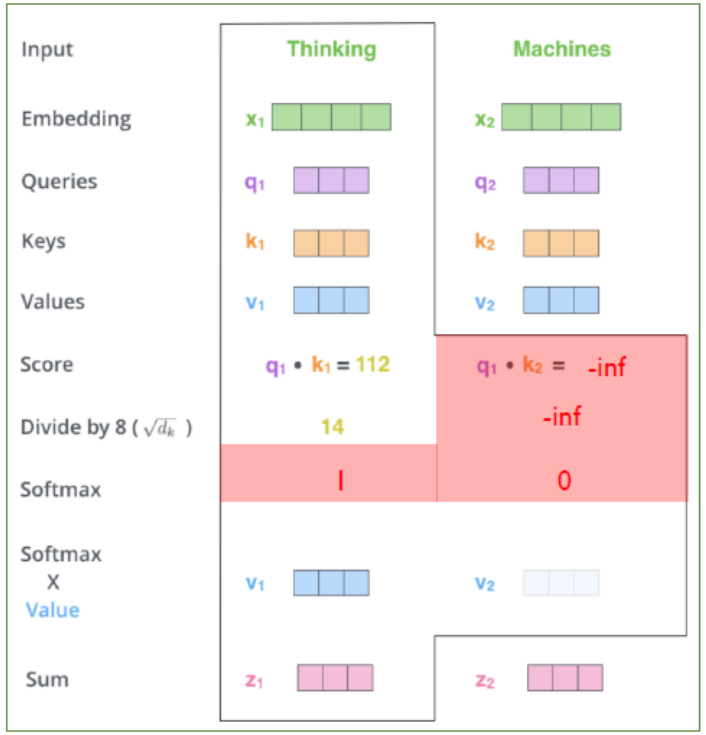




Transformer

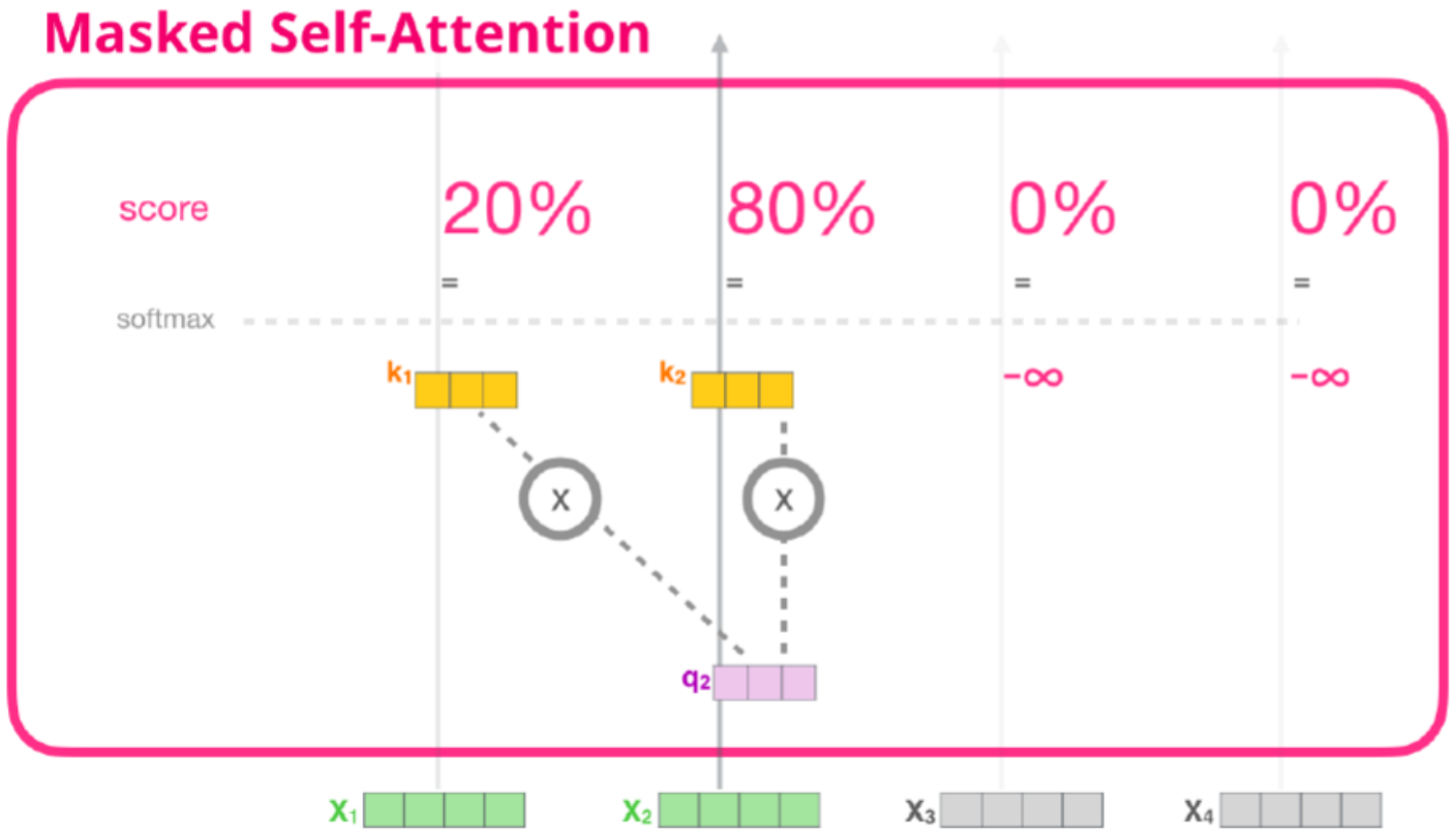
- Masked Multi Head Attention





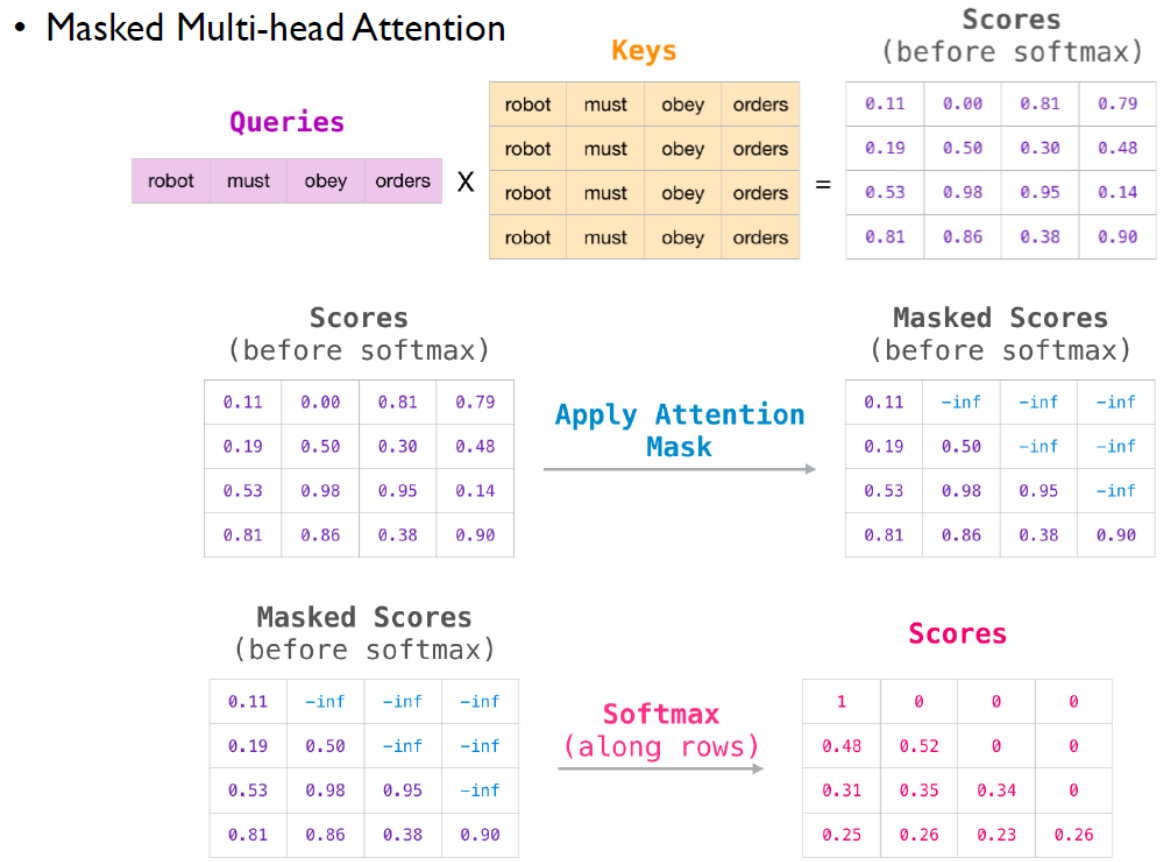
Transformer

- Masked Multi Head Attention



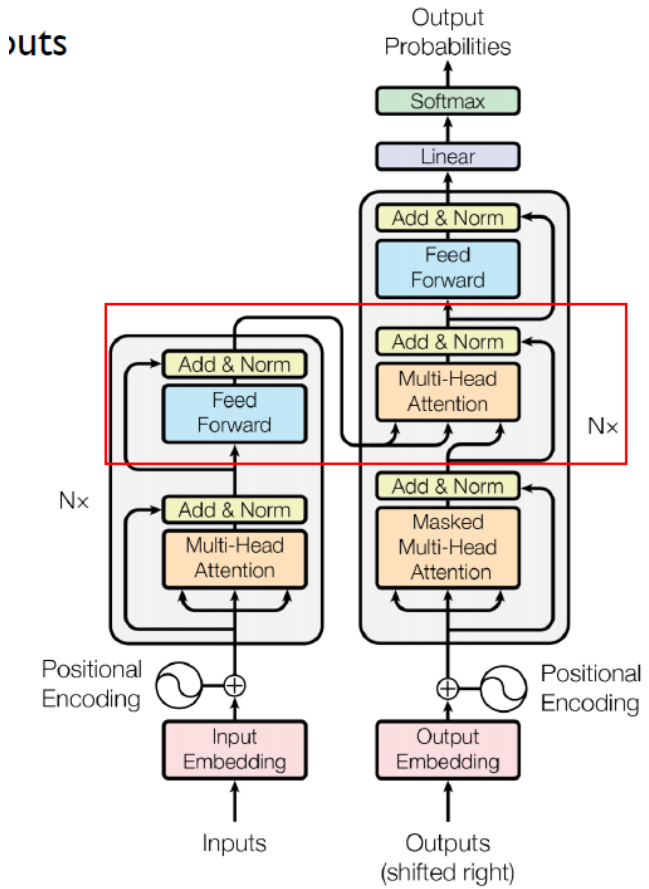
Transformer

Masked Multi Head Attention

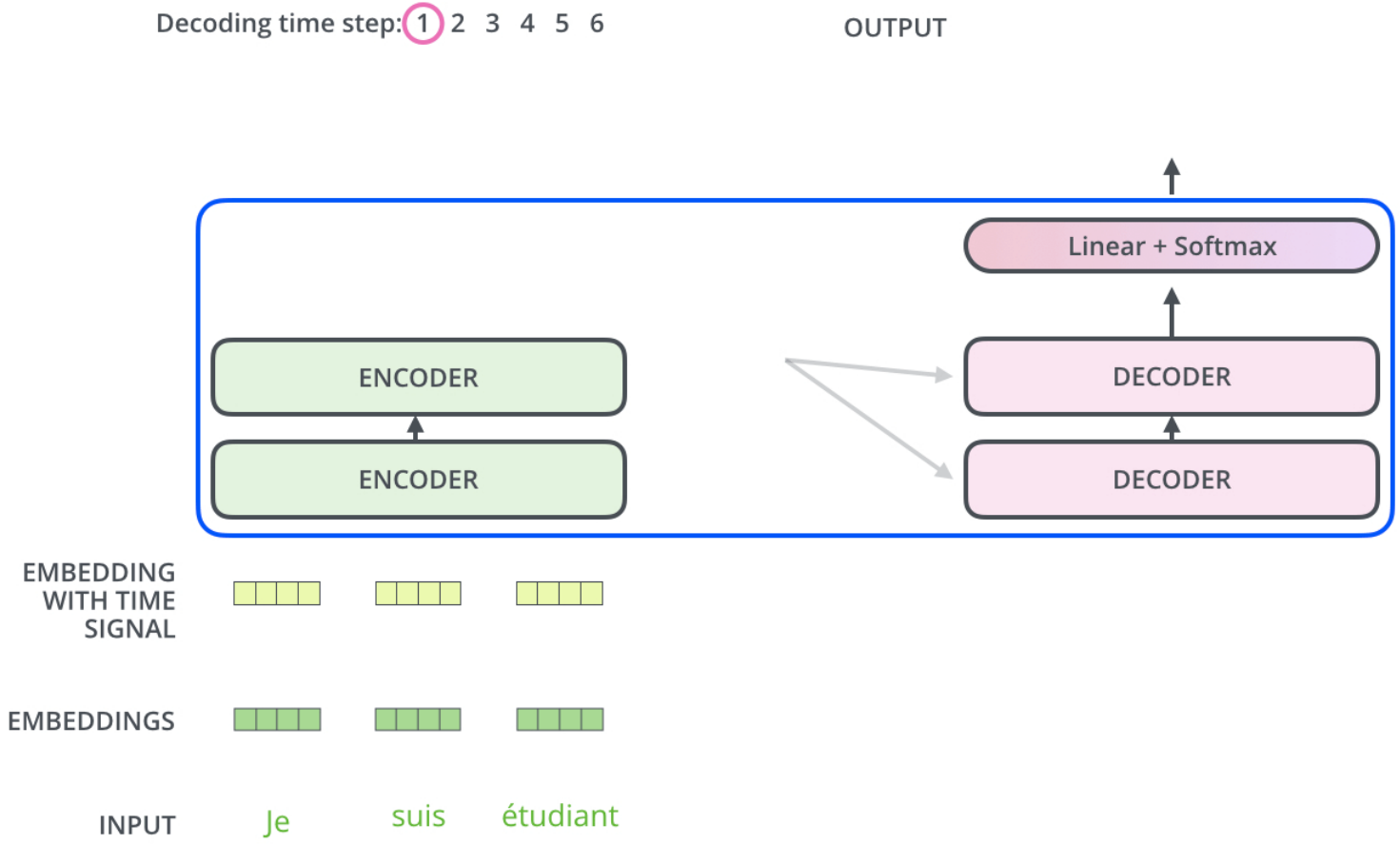


Transformer

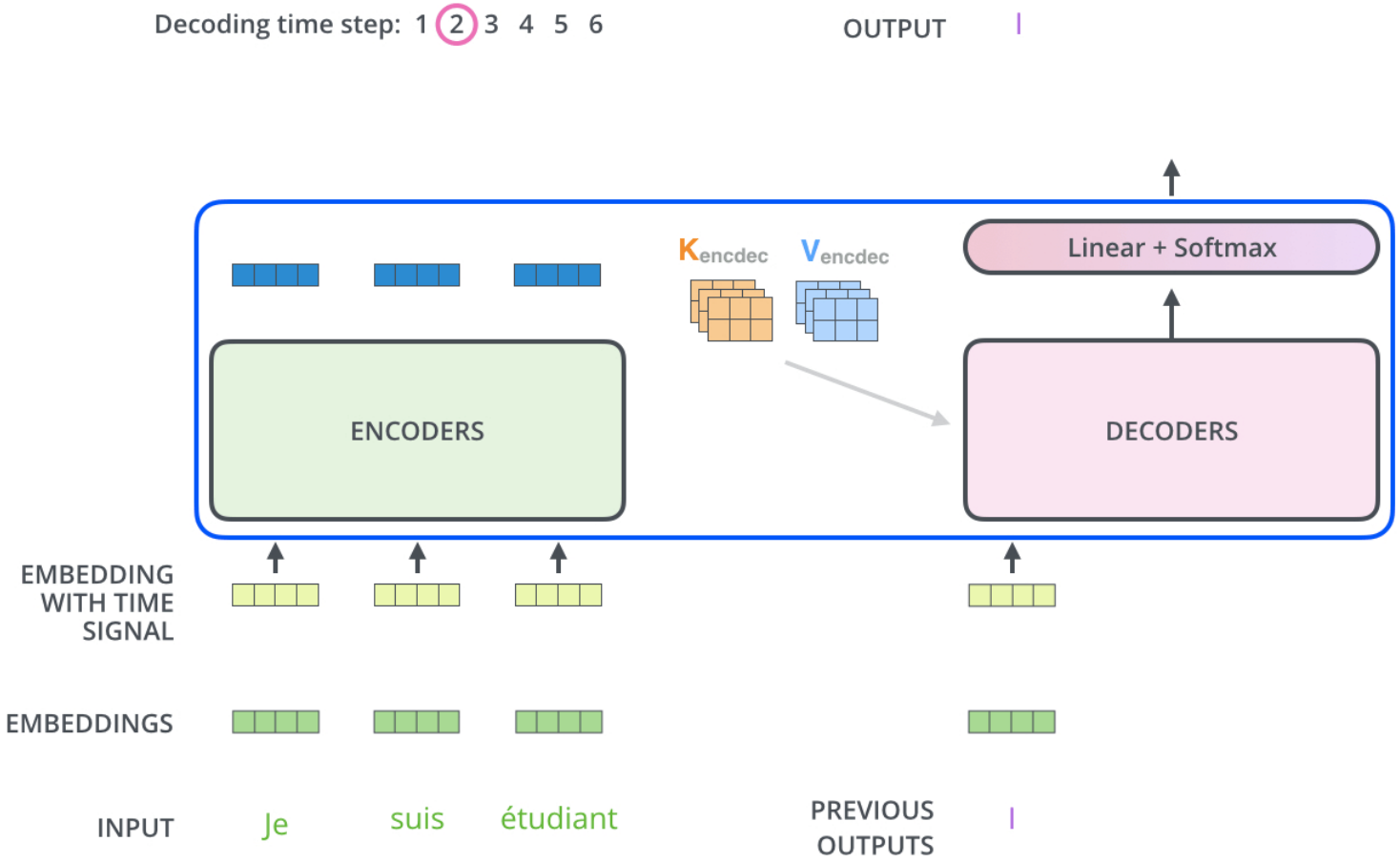
- Encoder Decoder Multi Attention



Transformer

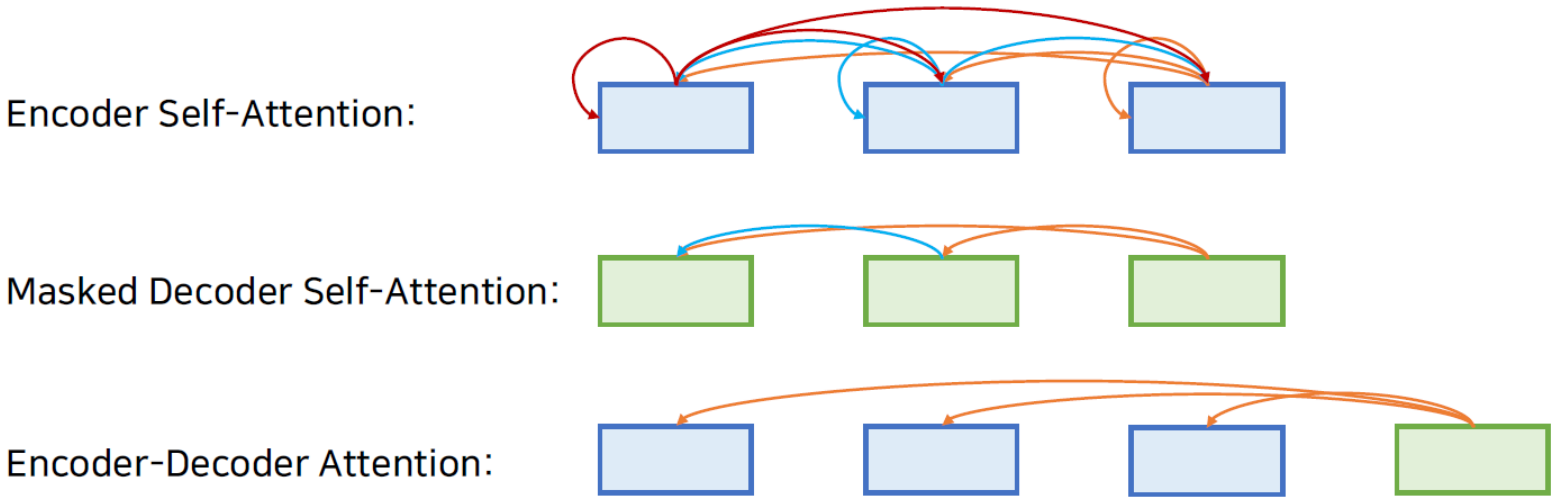


Transformer



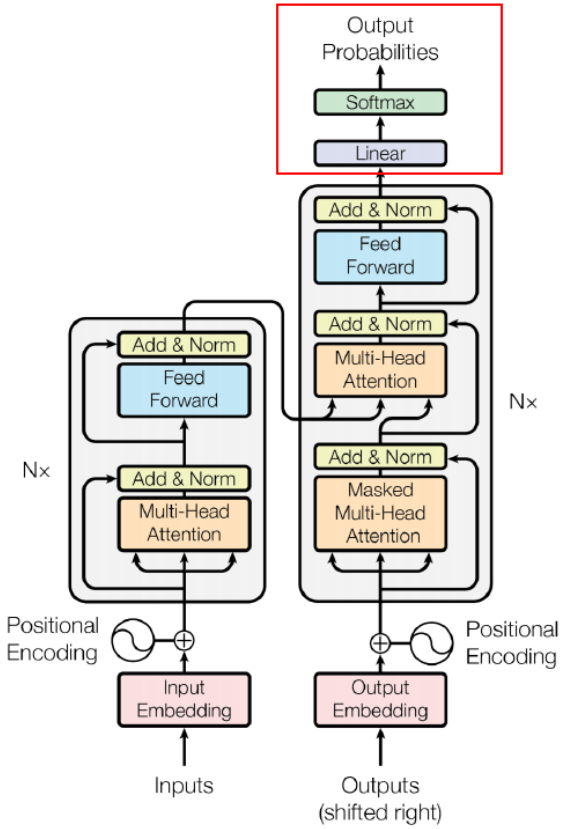
Transformer

- 3 Attention



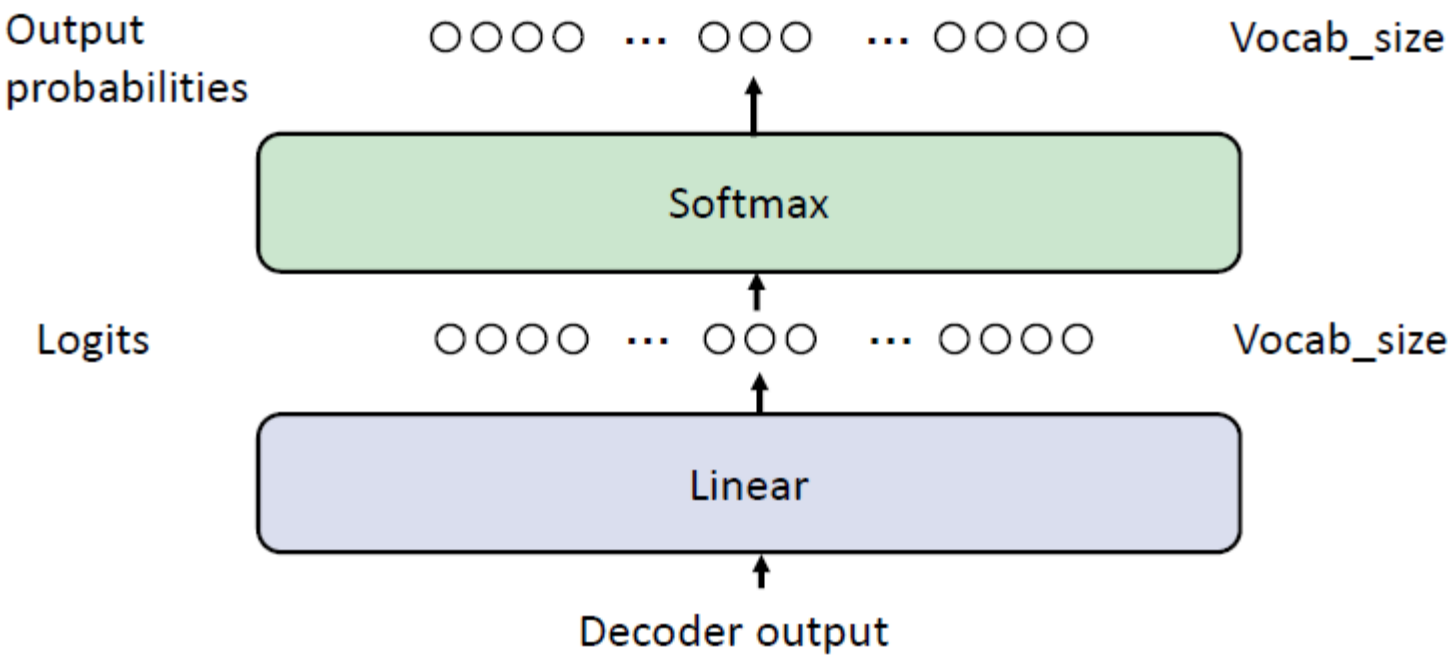
Transformer

- Output



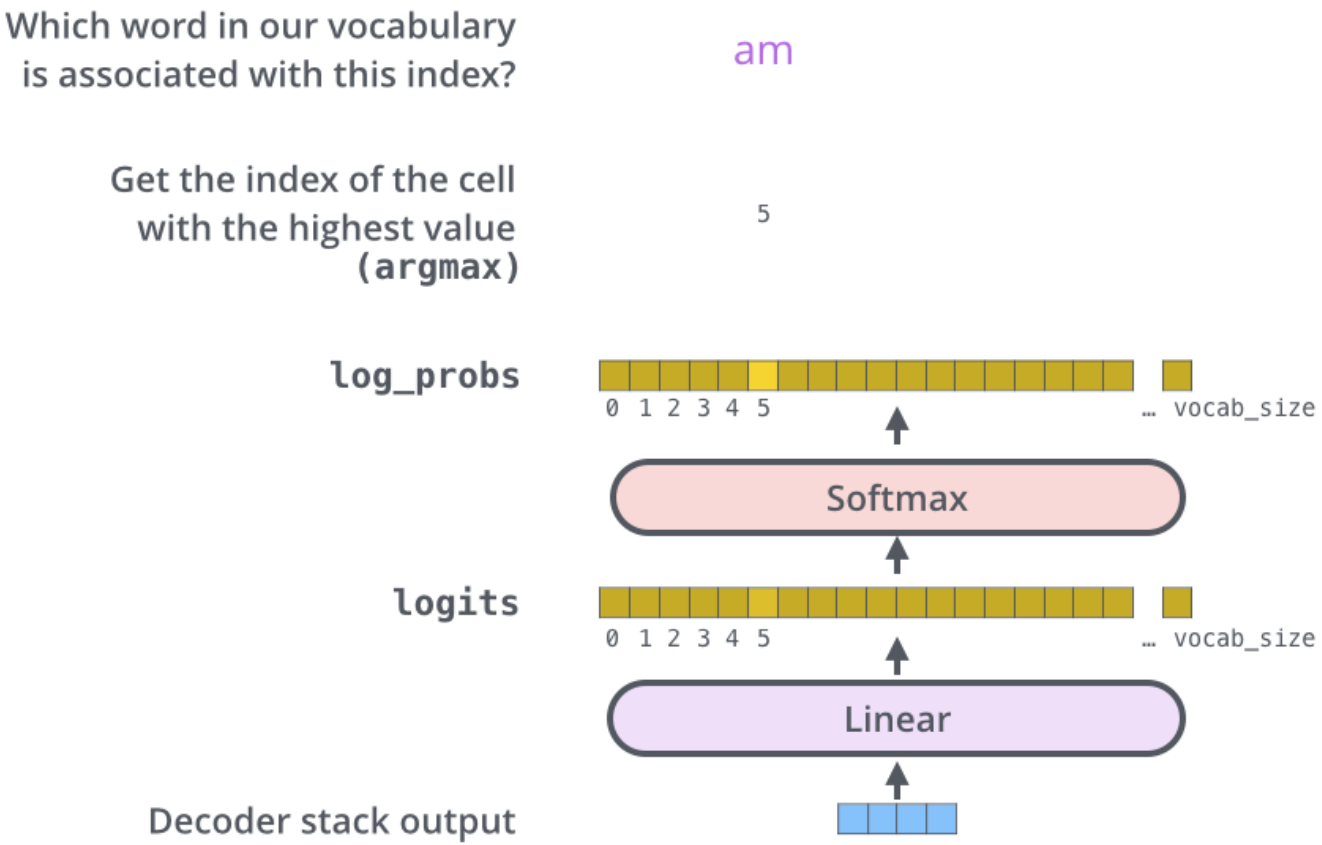
Transformer

- Output

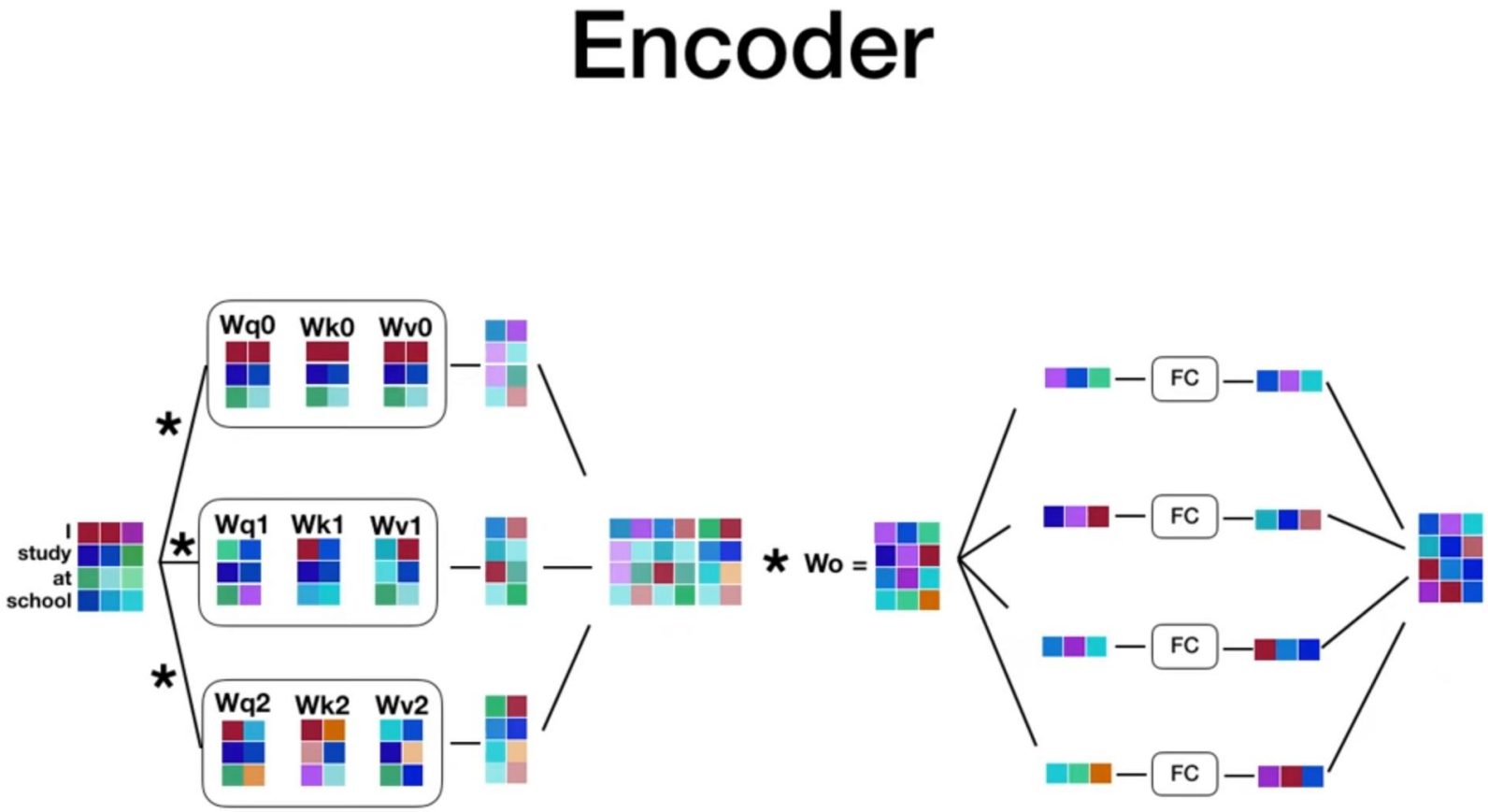


Transformer

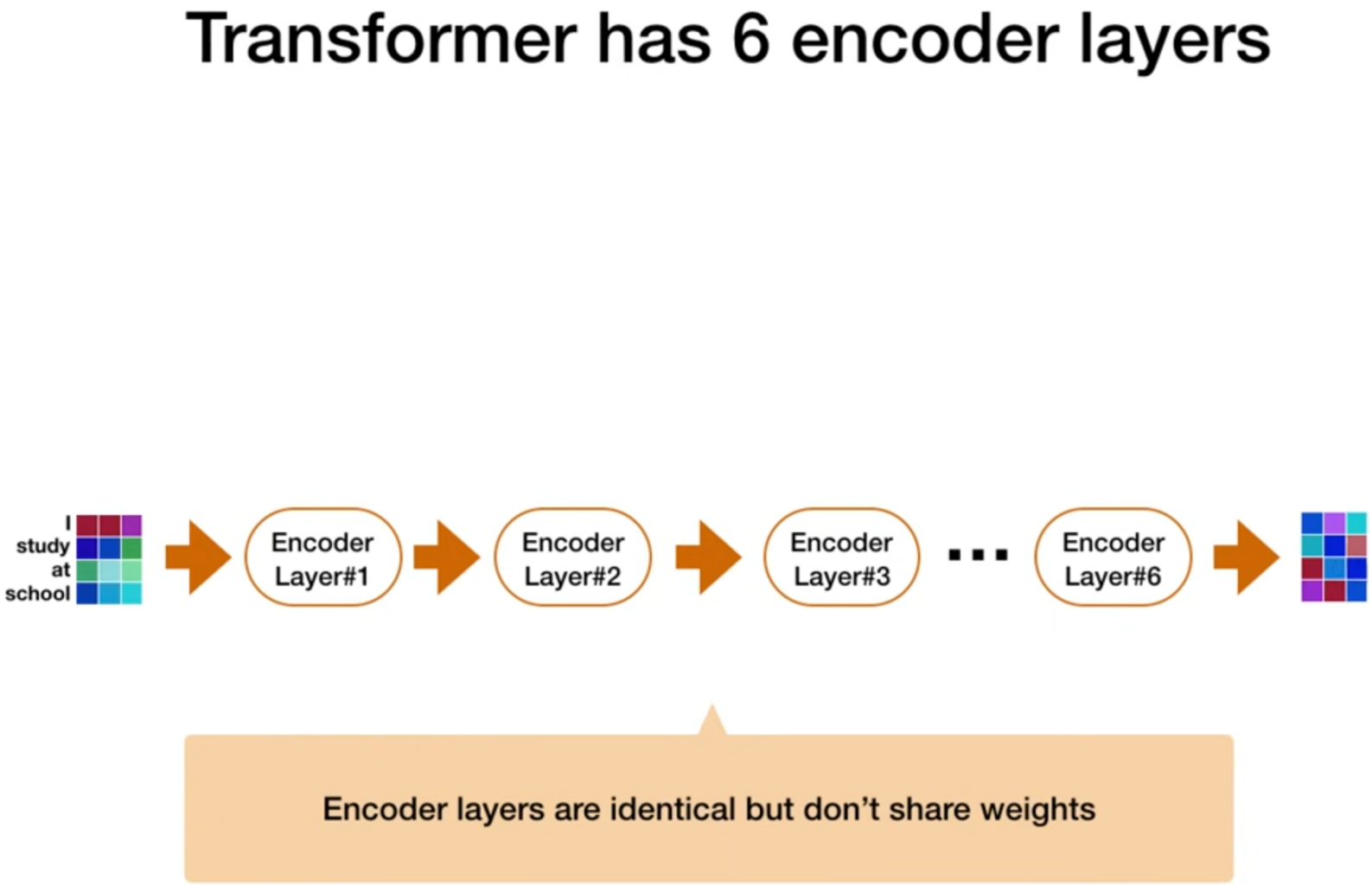
- Output



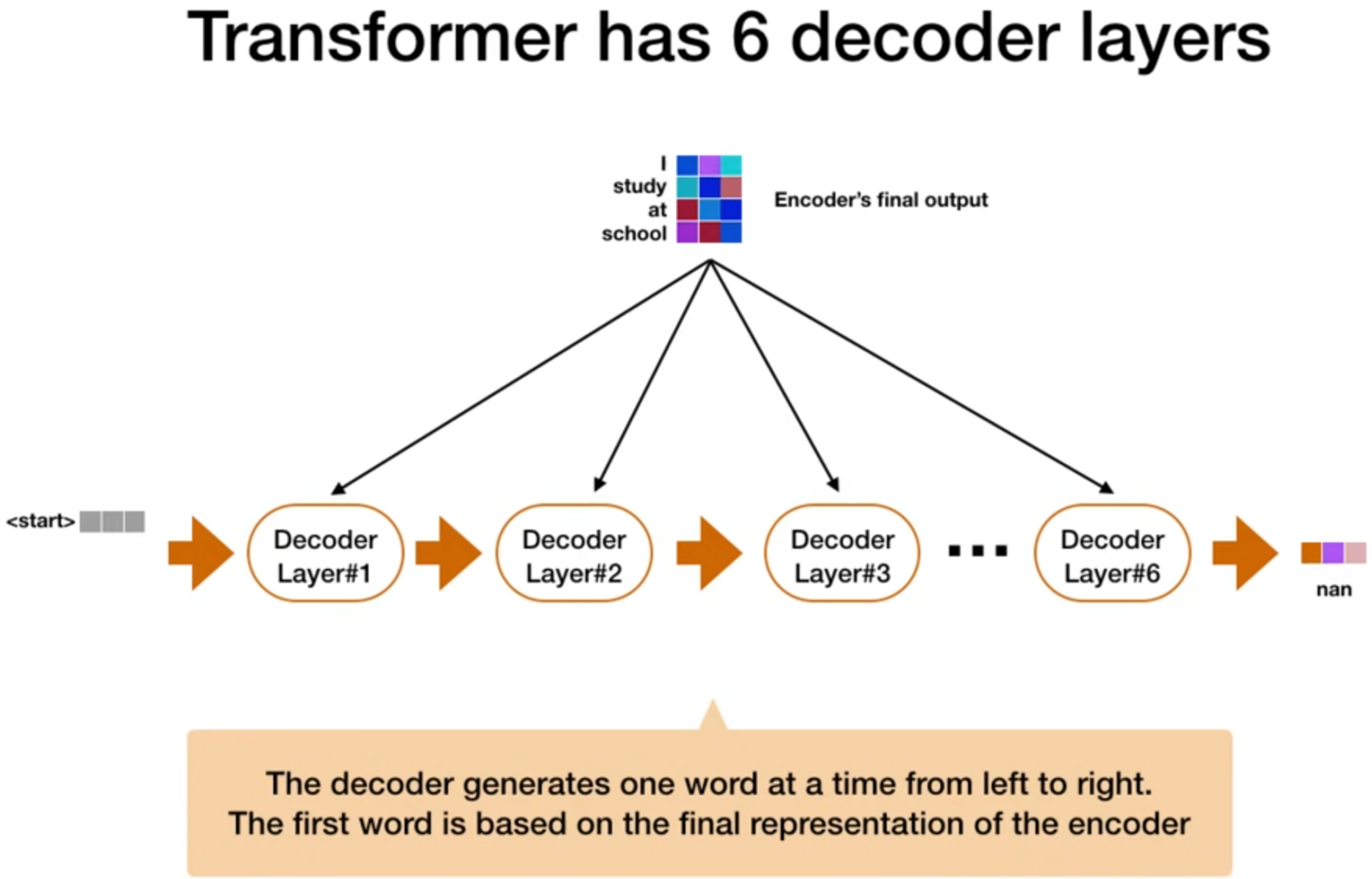
Summary


Encoder
Wq0
Wk0
Wv0
Wq1
Wk1
Wv1
Wq2
Wk2
Wv2
I
study
at
school
Wo =
FC
FC
FC
FC

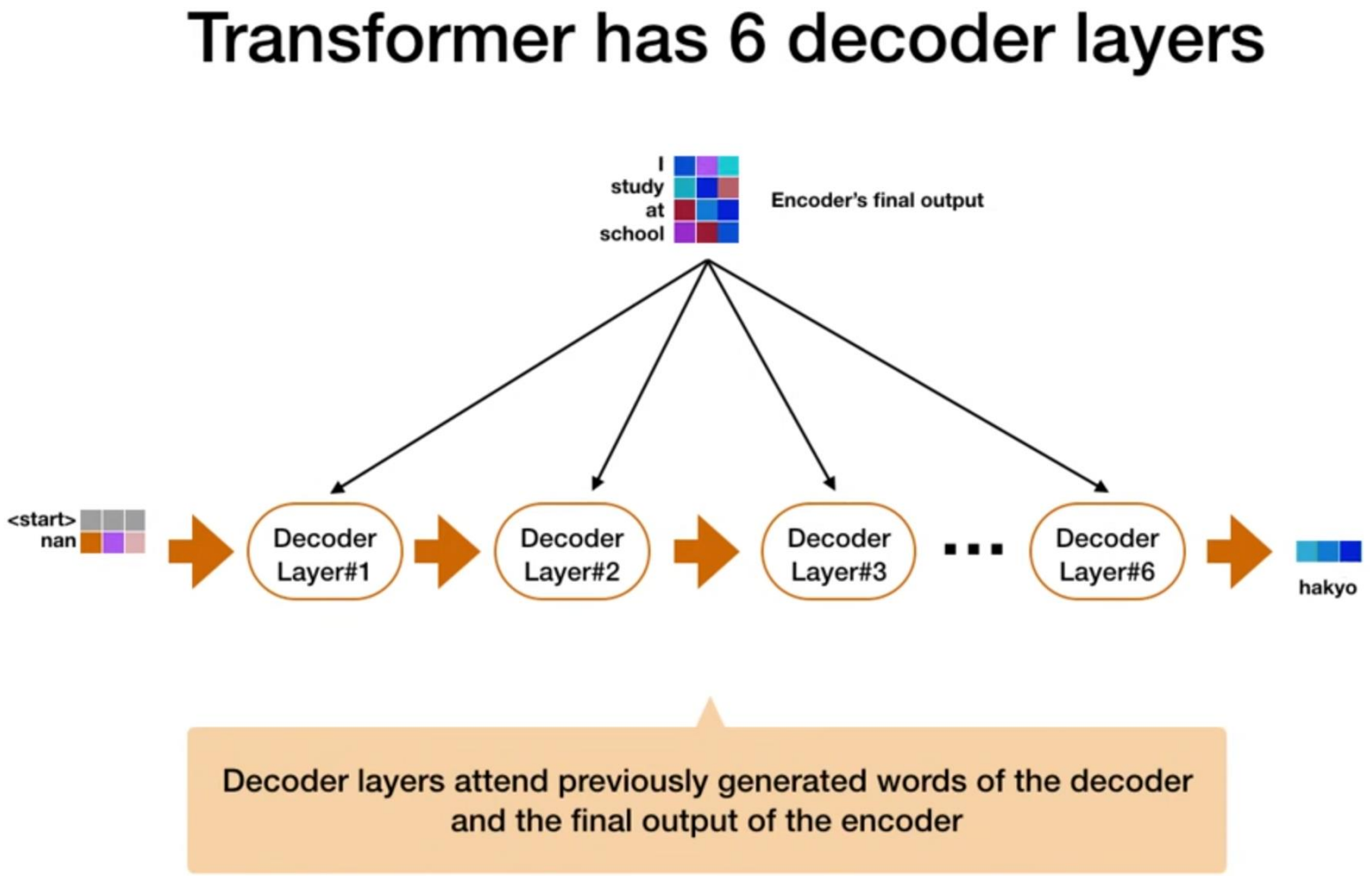
Summary


Transformer has 6 encoder layers
I
study
at
school
Encoder Layer#1
Encoder Layer#2
Encoder Layer#3
Encoder Layer#6
Encoder layers are identical but don’t share weights

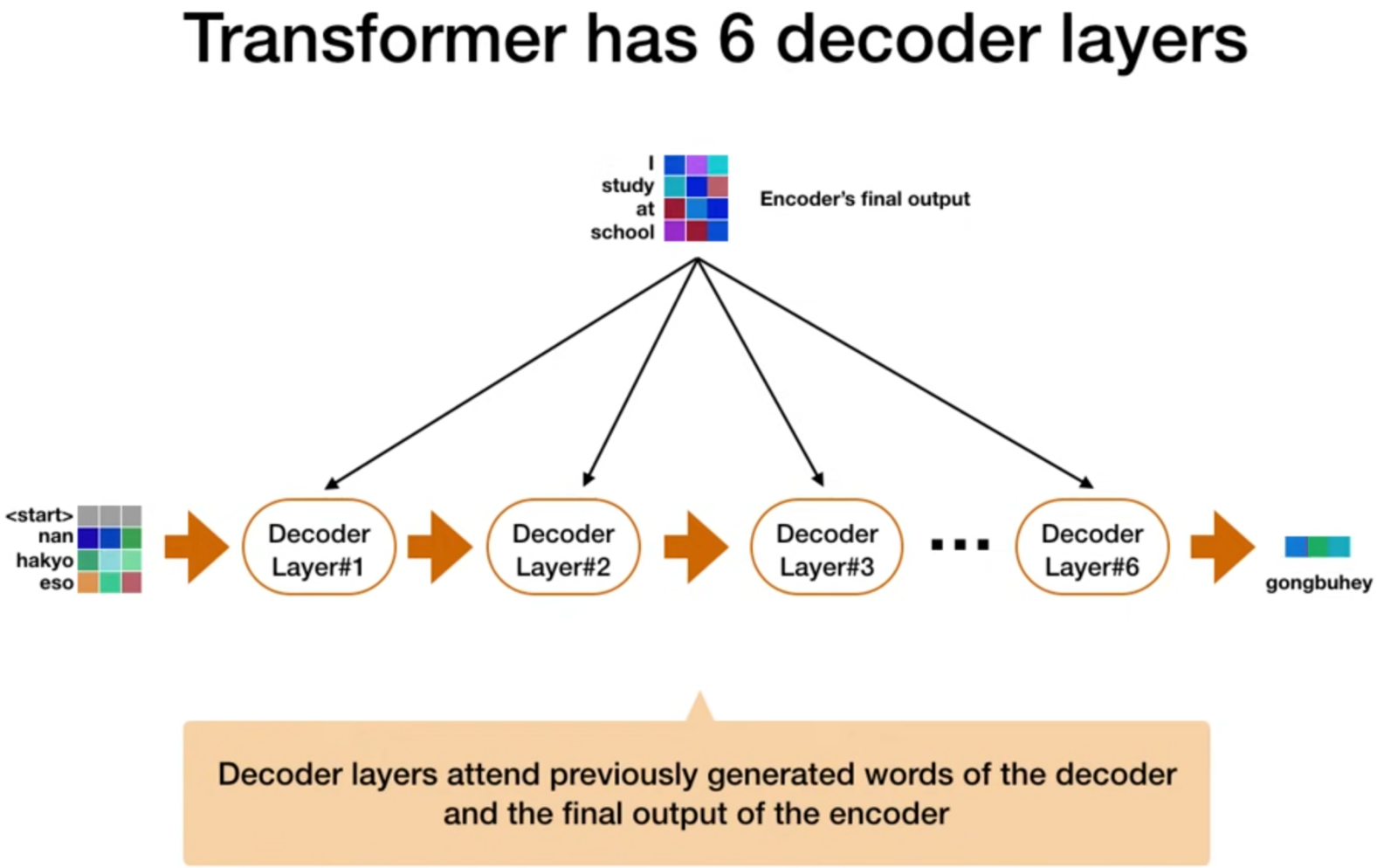
Summary


Transformer has 6 decoder layers
I
study
at
school
Encoder's final output
<start>
Decoder Layer#1
Decoder Layer#2
Decoder Layer#3
Decoder Layer#6
nan
The decoder generates one word at a time from left to right.
The first word is based on the final representation of the encoder

Summary


Transformer has 6 decoder layers
I
study
at
school
Encoder's final output
<start>
nan
Decoder Layer#1
Decoder Layer#2
Decoder Layer#3
Decoder Layer#6
hakyo
Decoder layers attend previously generated words of the decoder and the final output of the encoder

Summary


Transformer has 6 decoder layers
I
study
at
school
Encoder's final output
<start>
nan
hakyo
eso
Decoder Layer#1
Decoder Layer#2
Decoder Layer#3
Decoder Layer#6
gongbuhey
Decoder layers attend previously generated words of the decoder and the final output of the encoder

Transformer