Последовательность -> Последовательность

Машинный перевод

Пример:

ITA: Il gatto si e’ seduto sul tappetino.

EN: The cat sat on the mat.

Проблемы:

- Выравнивание: последовательности на входе / выходе могут иметь разную длину
- Неопределенность (отображение 1-ко-многим: множество возможных способов перевода)
- Метрика: как автоматически оценивать, означают ли предложения к одно и то же?

Последовательность -> Последовательность

Машинный перевод

Пример:

ITA: Il gatto si e’ seduto sul tappetino.

EN: The cat sat on the mat.

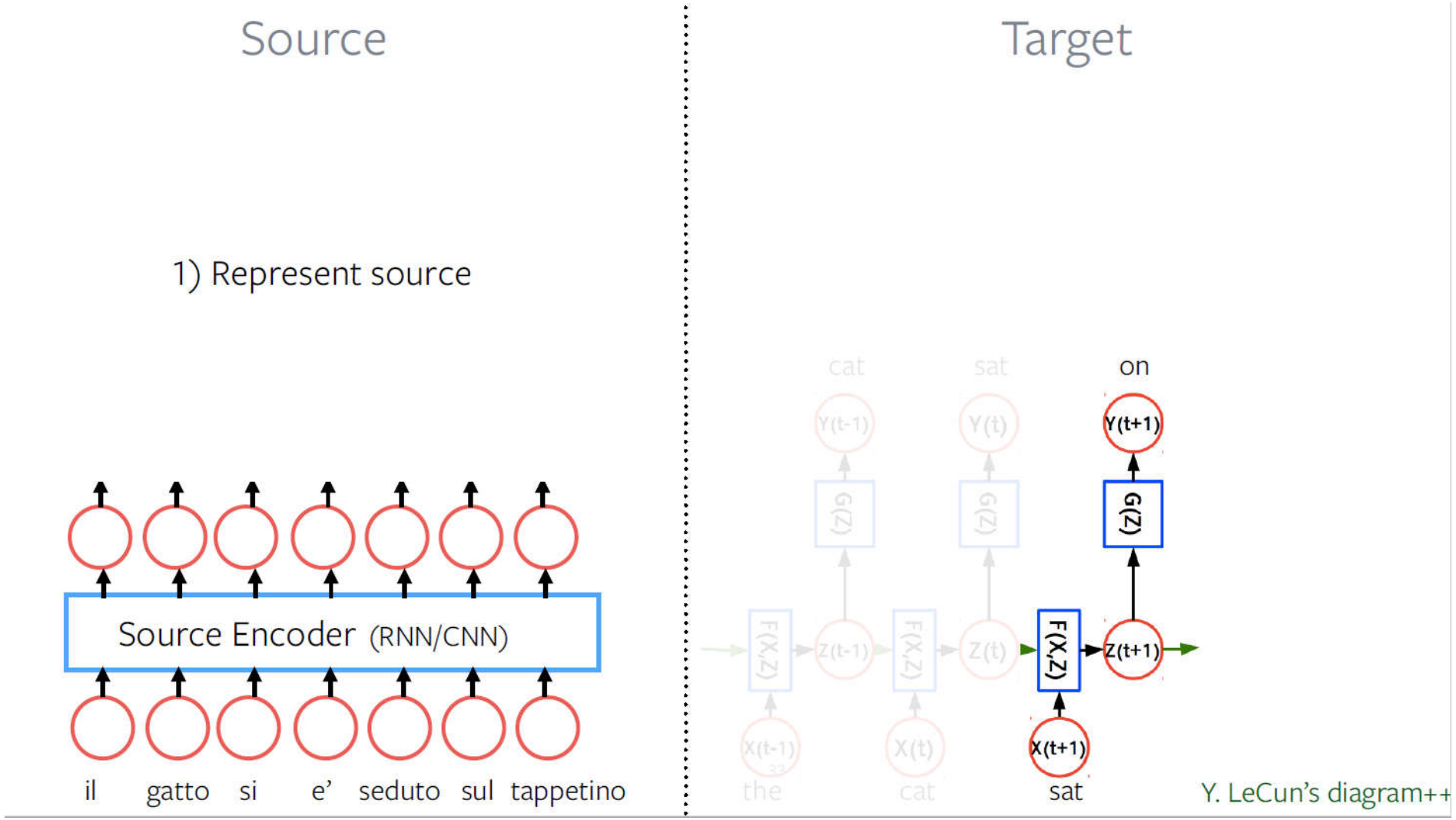
Подход:

Имеем один RNN для кодирования исходного предложения, а другой RNN - для предсказания целевого предложения. Целевой RNN учится (мягко) выравнивать предложение с помощью механизма внимания.

Neural machine translation by jointly learning to align and translate, Bahdanau et al. ICLR 2015

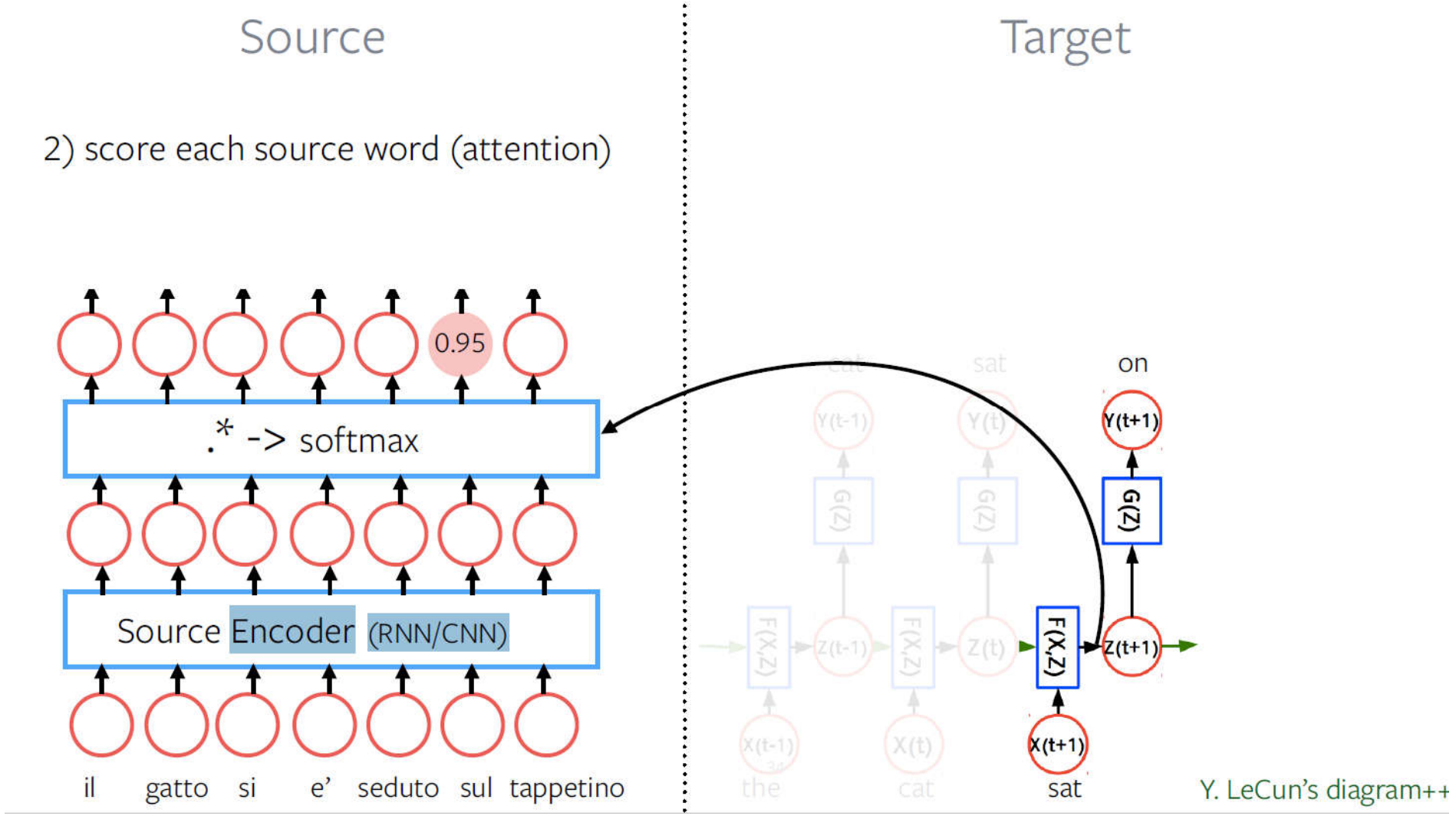
Последовательность -> Последовательность

Машинный перевод



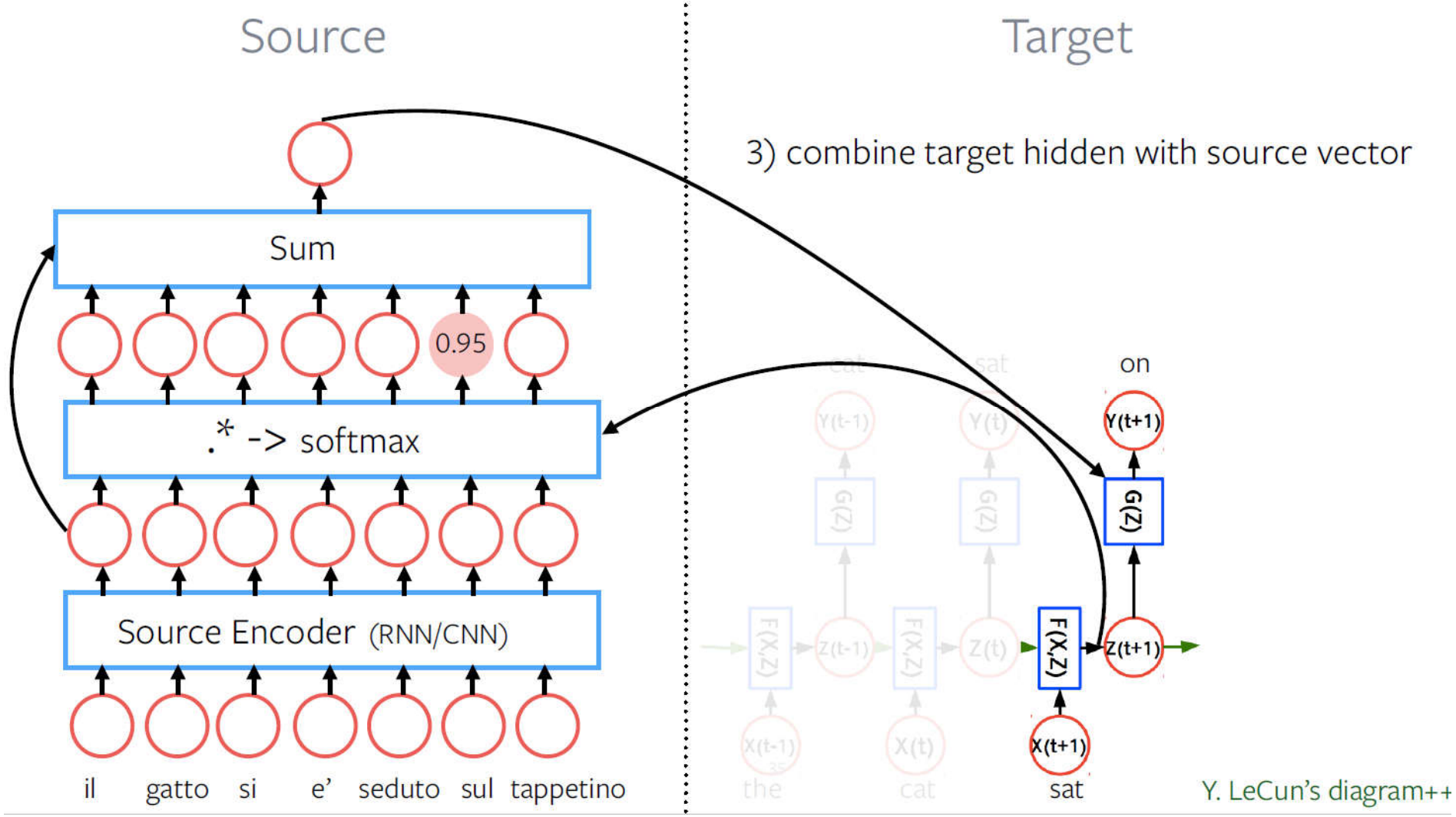
Последовательность -> Последовательность

Машинный перевод



Последовательность -> Последовательность

Машинный перевод



Последовательность -> Последовательность

Машинный перевод

Пример:

ITA: Il gatto si e’ seduto sul tappetino.

EN: The cat sat on the mat.

Заметки:

+ Исходное и целевое предложение может иметь любую длину, хорошо работает и при длинных предложениях!
+ Учится неявно выравнивать
+ RNN можно заменить на CHC. *A convolutional encoder model for NMT, Gehring et al. 2016*
+ Генерирует беглые предложения
- Имеет проблемы с редко встречаемыми словами, точным выбором слов
- Обычно обучается как языковая модель (кросс-энтропия), хороша для вычисления оценки, но не для генерации

Последовательность -> Последовательность

Машинный перевод

Заключение:

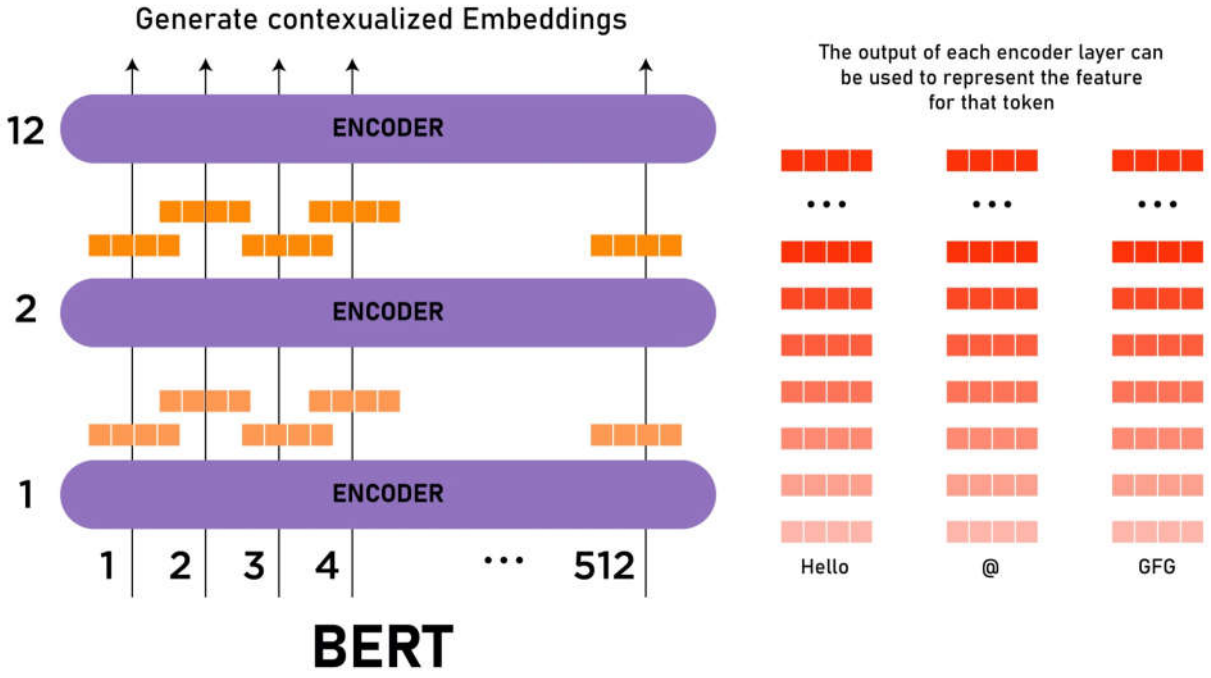
+ Механизм внимания является довольно общим и может использоваться для:
    + Работает с входами переменной длины, поскольку «мягко выбирает один»
    + Неявное выравнивание, которое обучается моделью, по мере необходимости
    + Для выполнения раундов «объяснений» (например, «переходов» в сетях памяти)
+ Этот же механизм использовался для создания субтитров, суммирования и т. Д.

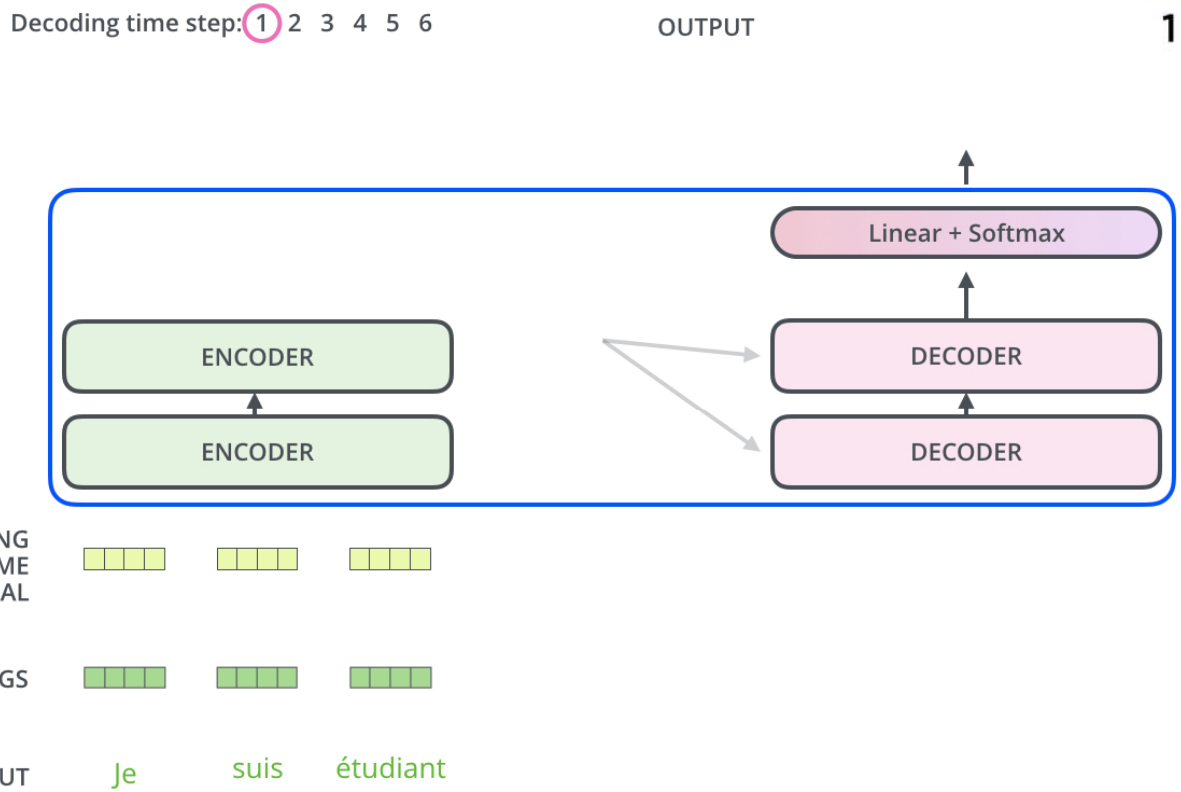
- Функция потери на уровне слов (кросс-энтропия для предсказывания следующего слова) является субоптимальной для задач генерации текстов

Sequence level training with RNNs, Ranzato et al. ICLR 2016
An actor-critic algorithm for sequence prediction, ICLR 2017
Sequence-to-sequence learning as beam-search optimization, EMNLP 2016

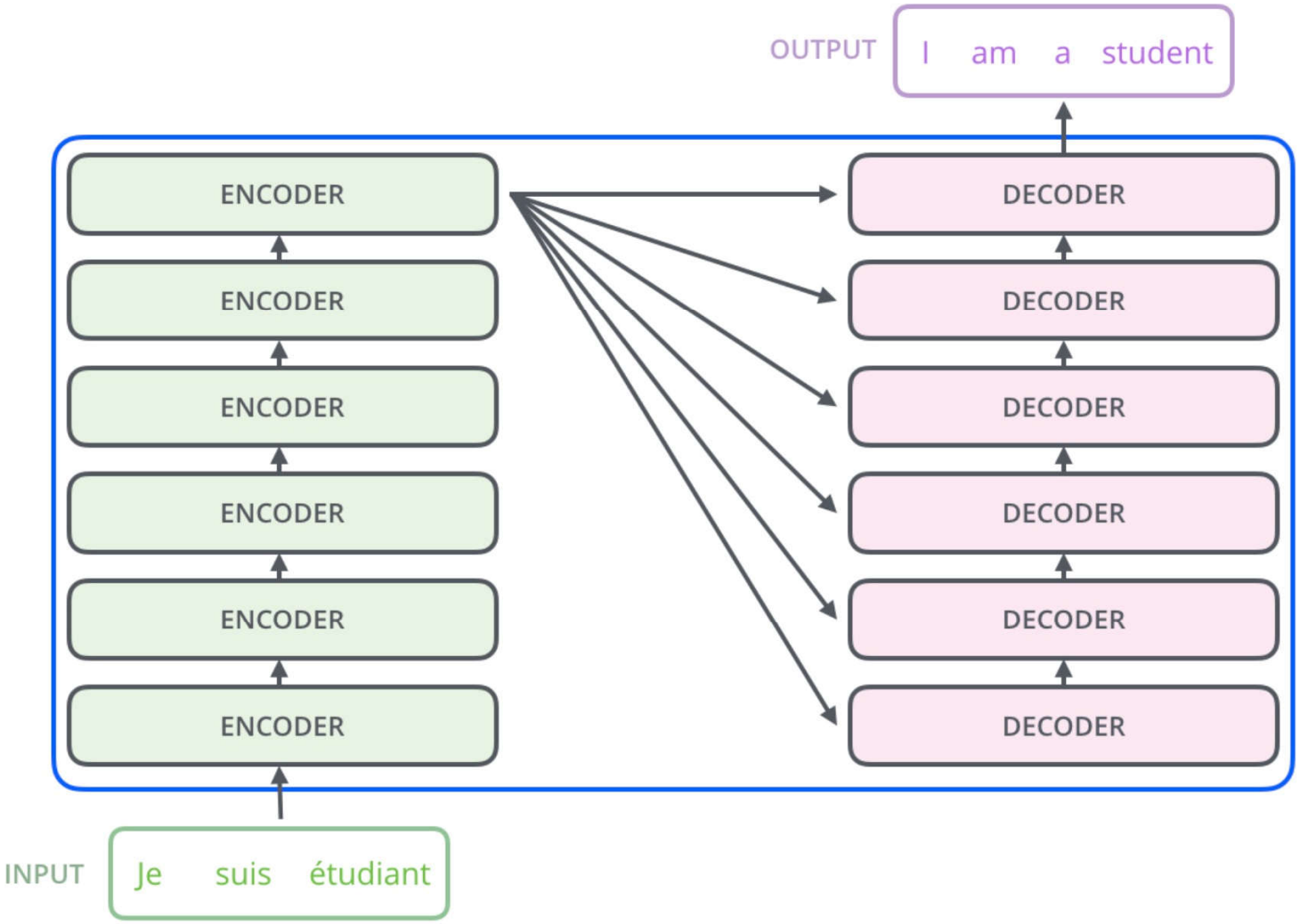
Трансформер

- Трансформер с механизмом внимания
- BERT, GPT

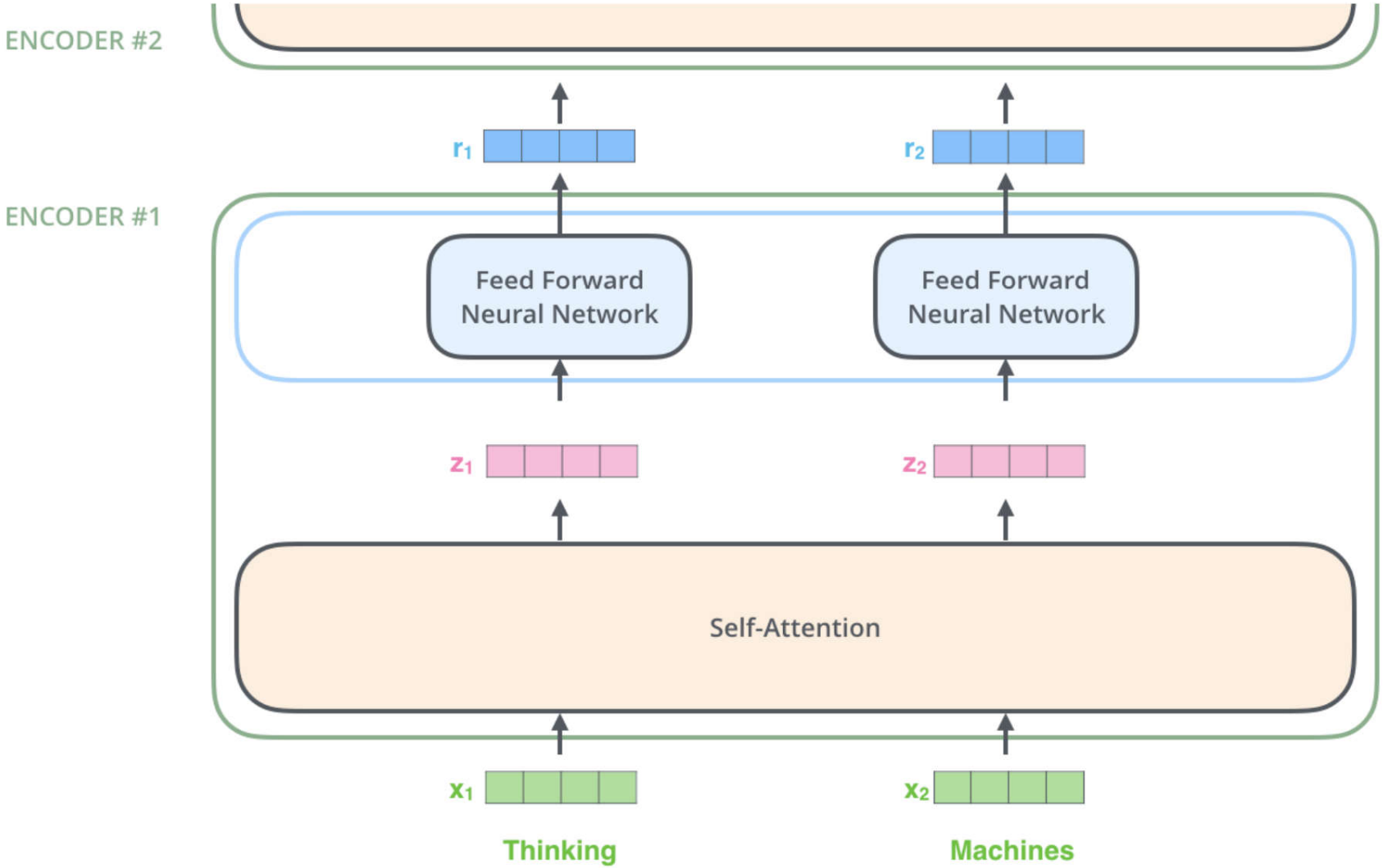




Архитектура Transformer



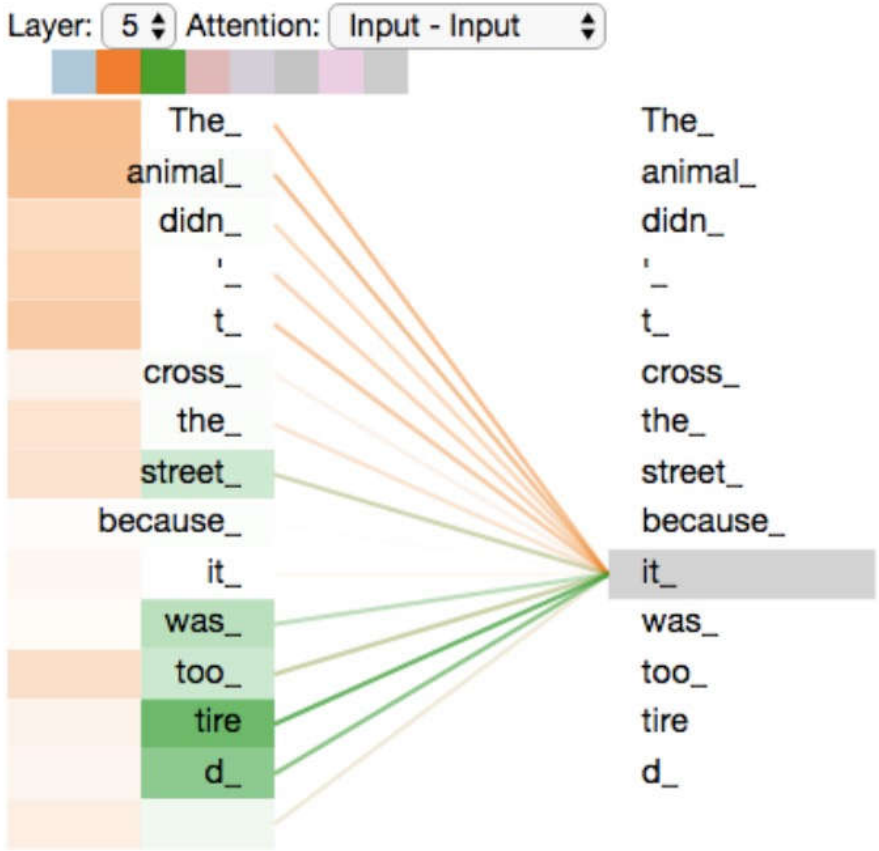
Кодировщик


ENCODER #2
ENCODER #1
r1
r2
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Self-Attention
x1
x2
Thinking
Machines

Внутреннее внимание

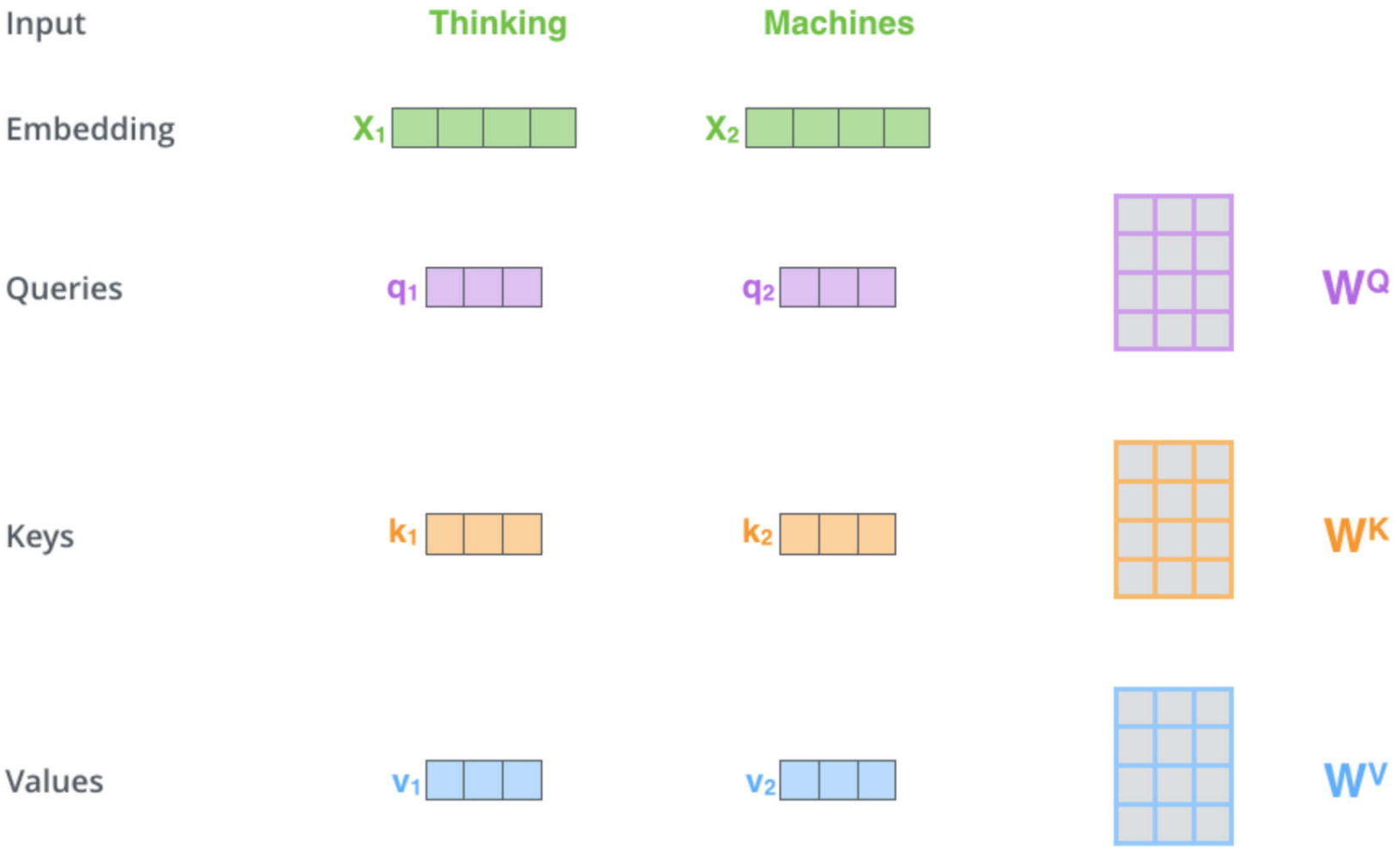
As we encode the word "it", one attention head is focusing most on "the animal", while another is focusing on "tired" -- in a sense.

The model's representation of the word "it" bakes in some of the representation of both "animal" and "tired"

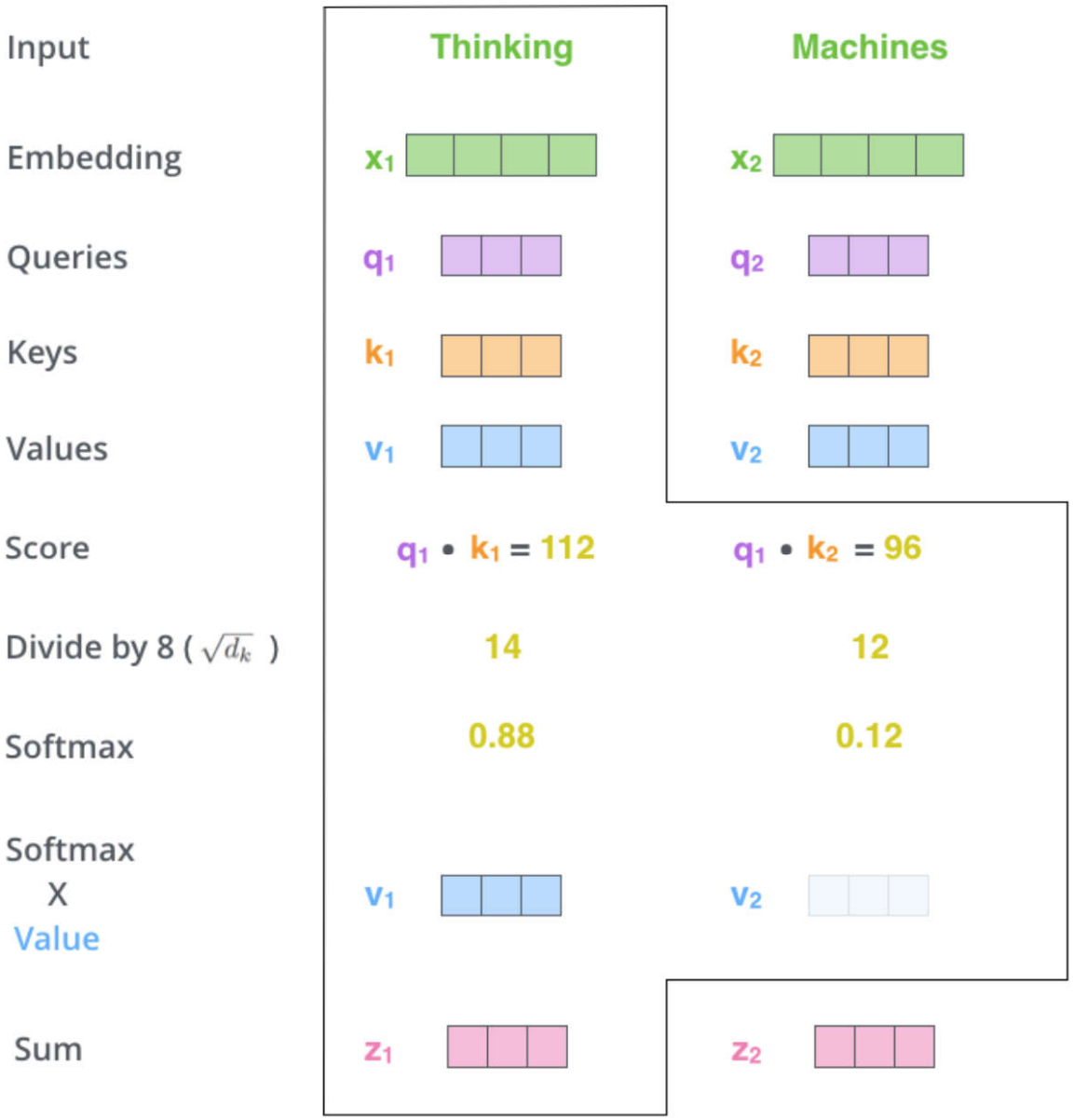


Внутреннее внимание

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.



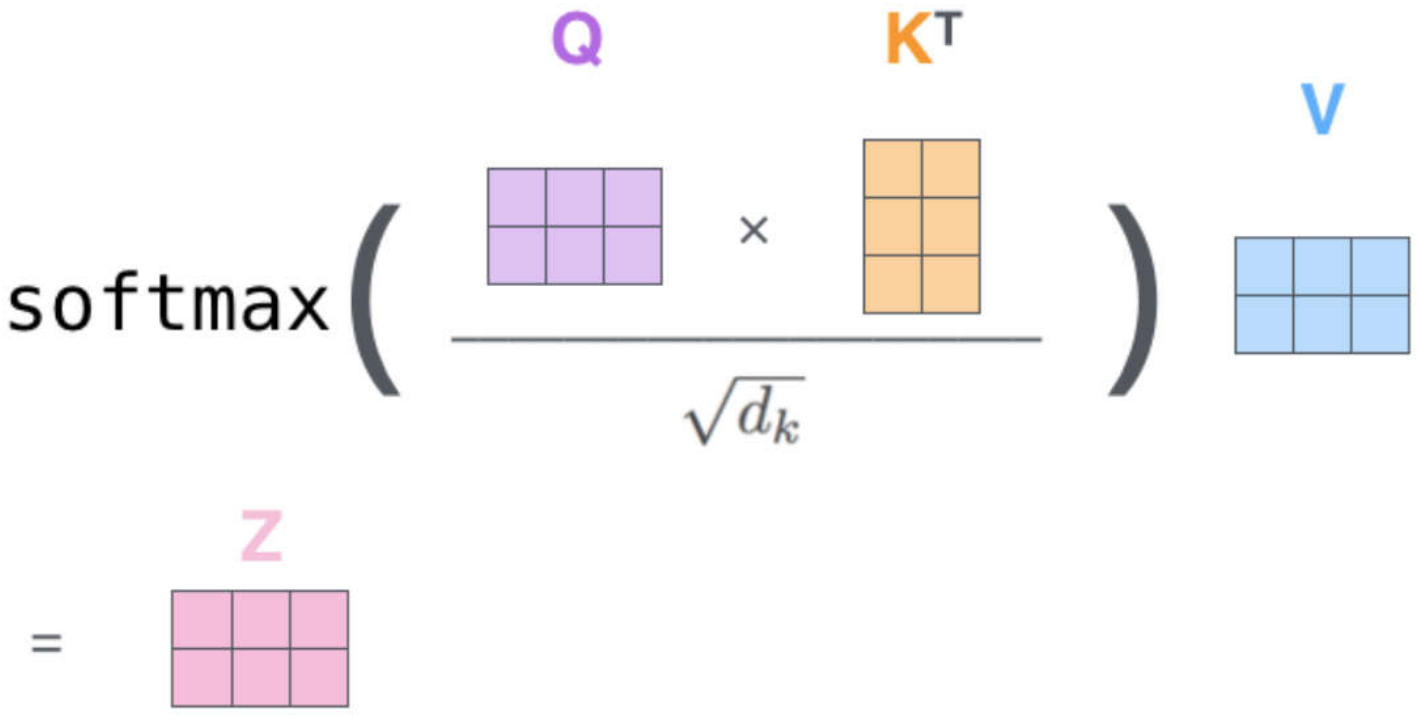
Value


Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 (√dk)
14
12
Softmax
0.88
0.12
Softmax
X
Value
v1
v2
Sum
z1
z2

Several words

Every row in the X matrix corresponds to a word in the input sentence

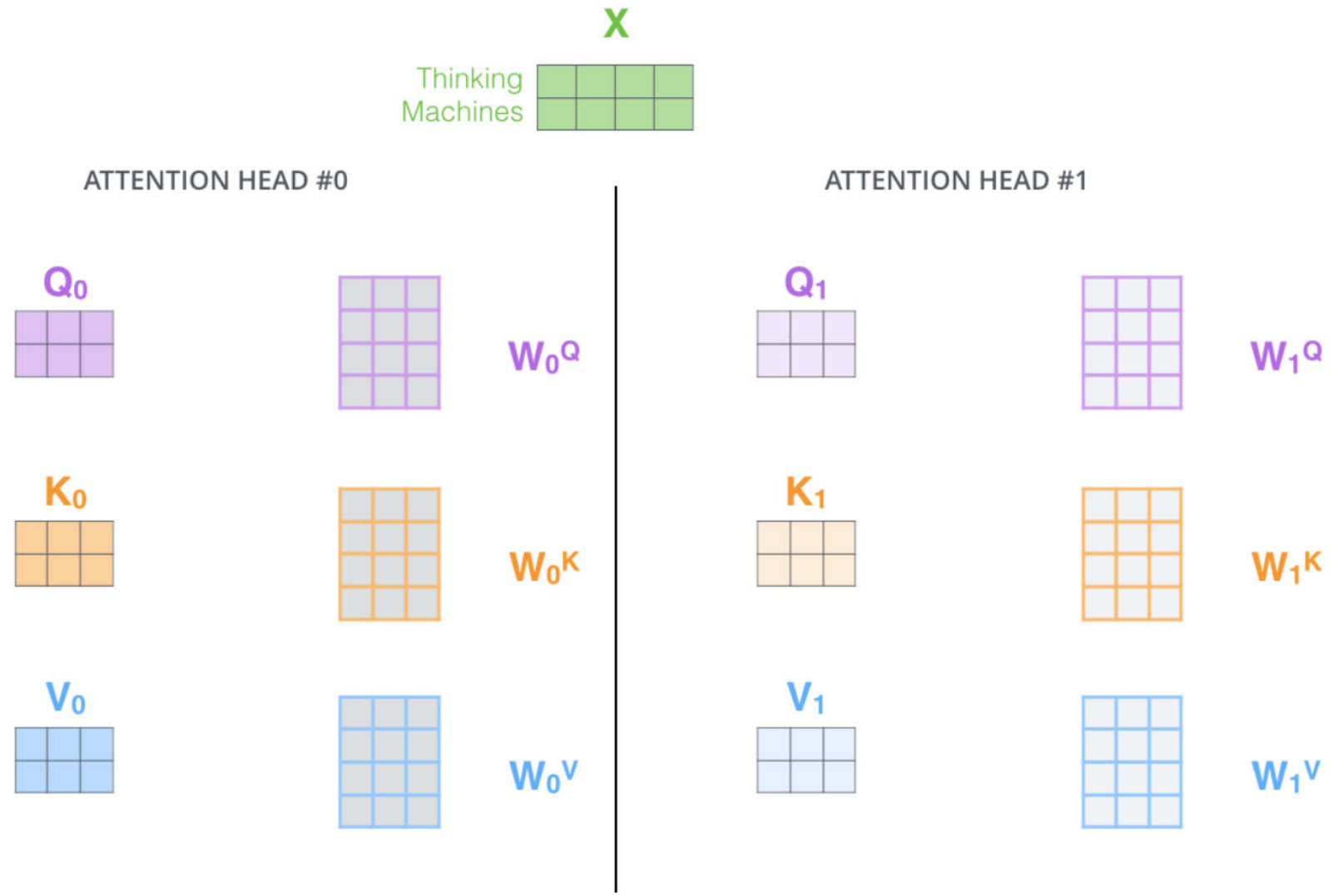
Calculation



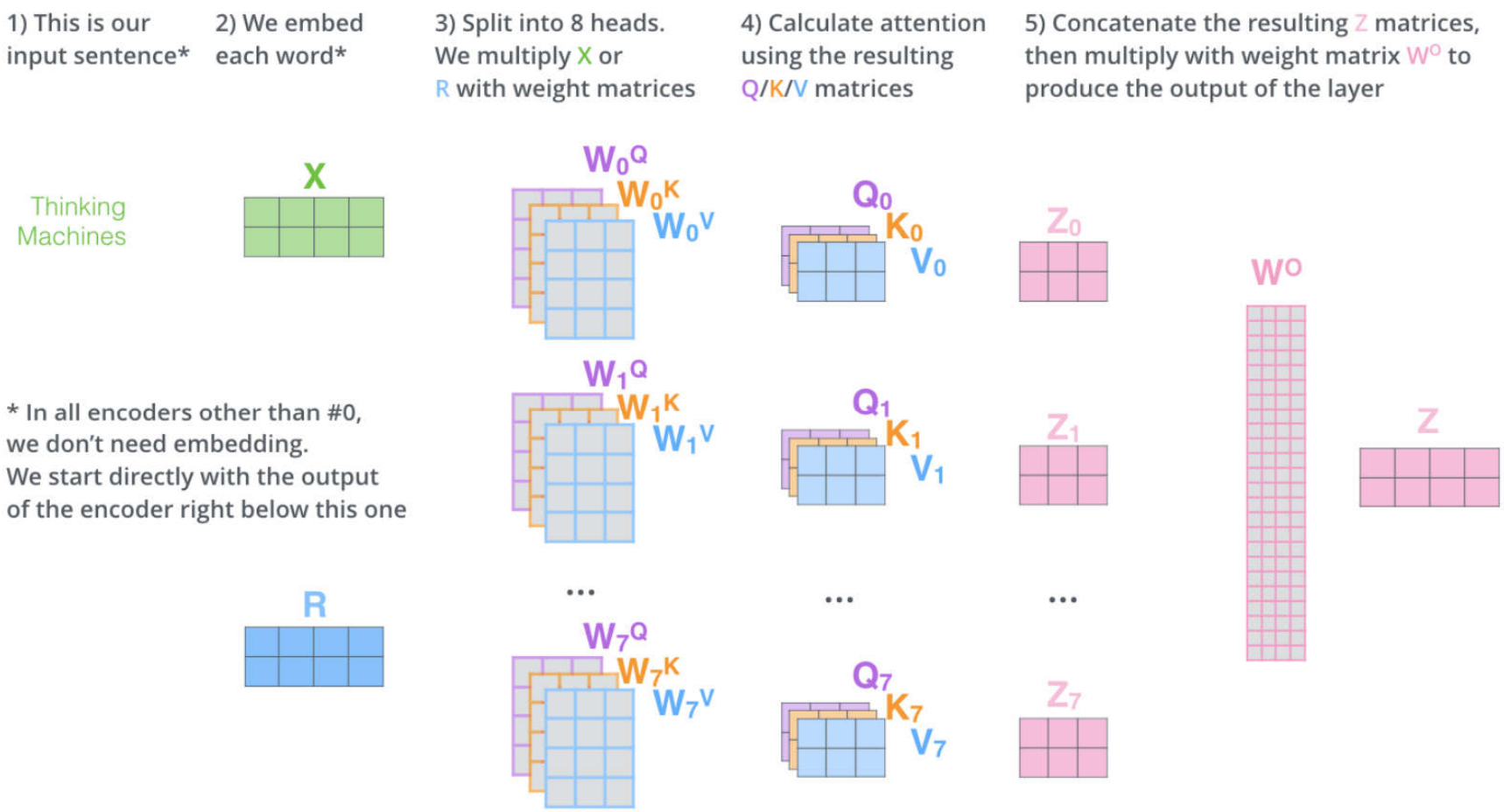
The self-attention calculation in matrix form

Attention heads

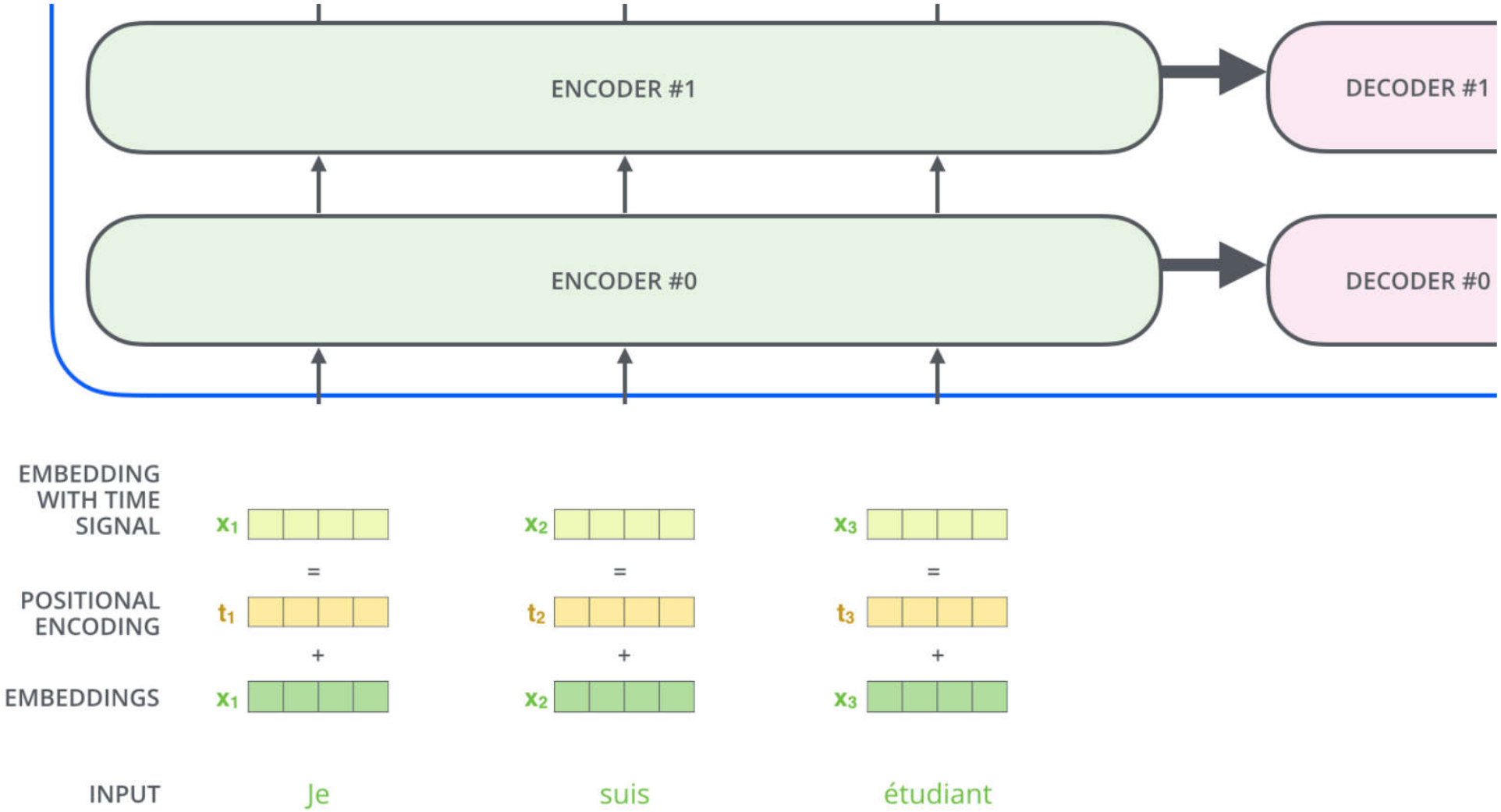
With multi-headed attention, we maintain separate Q/K/V weight matrices for each head resulting in different Q/K/V matrices. As we did before, we multiply X by the WQ/WK/WV matrices to produce Q/K/V matrices



Attention result



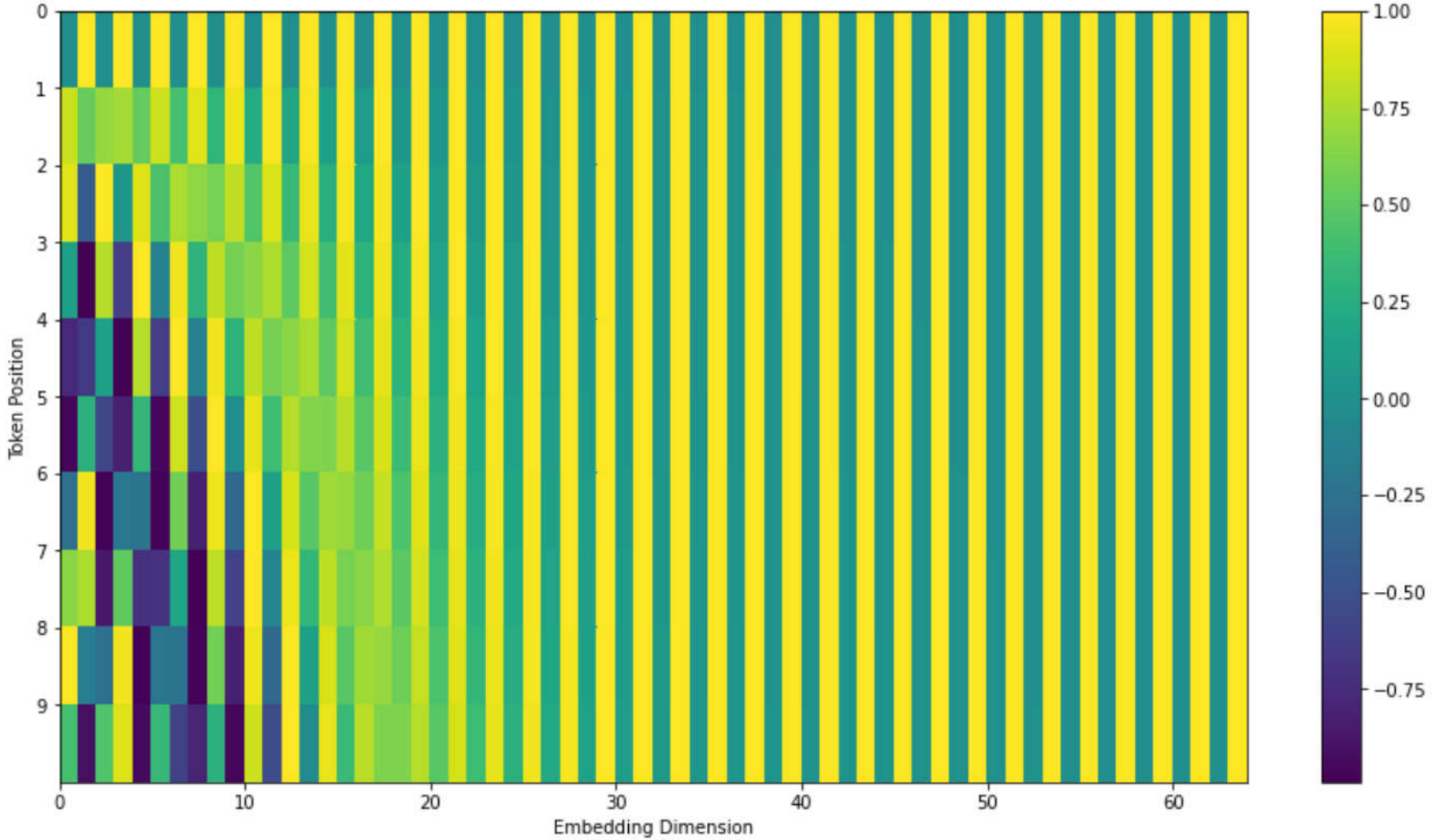
Positions


ENCODER #1
DECODER #1
ENCODER #0
DECODER #0
EMBEDDING WITH TIME SIGNAL
POSITIONAL ENCODING
EMBEDDINGS
INPUT
x1
x2
x3
t1
t2
t3
=
+
Je
suis
étudiant

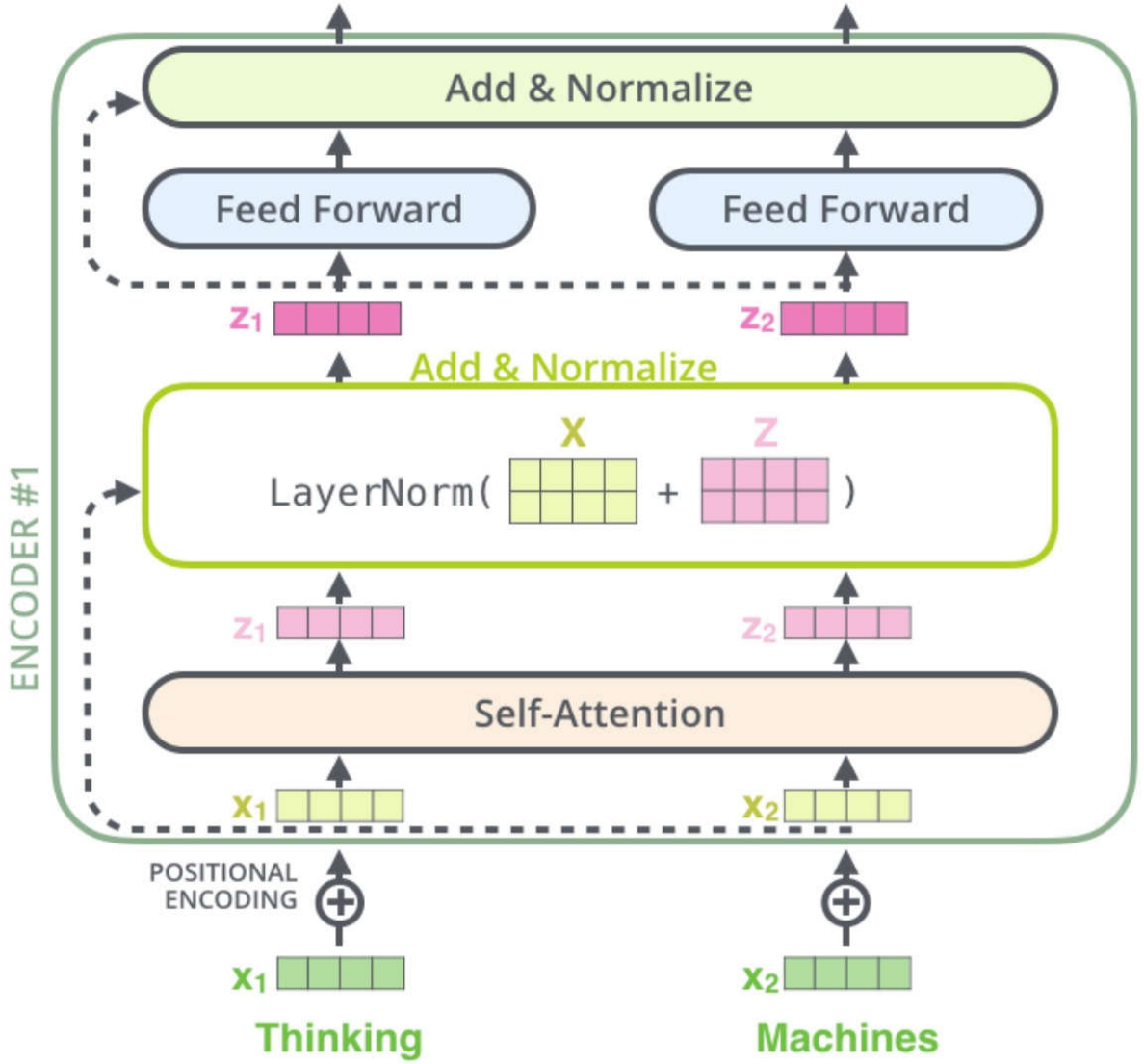
Positional encoding

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

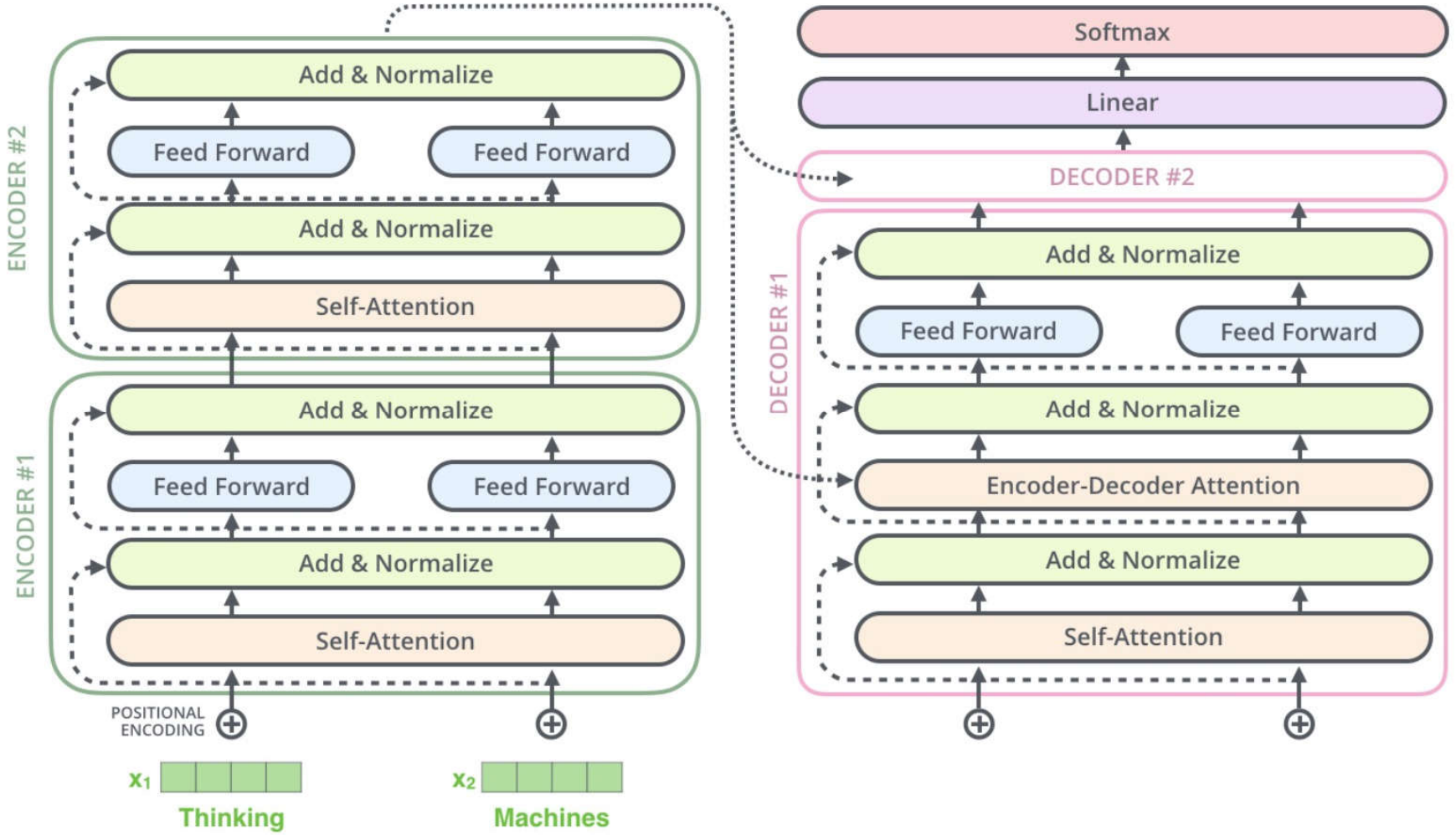
Positional encoding



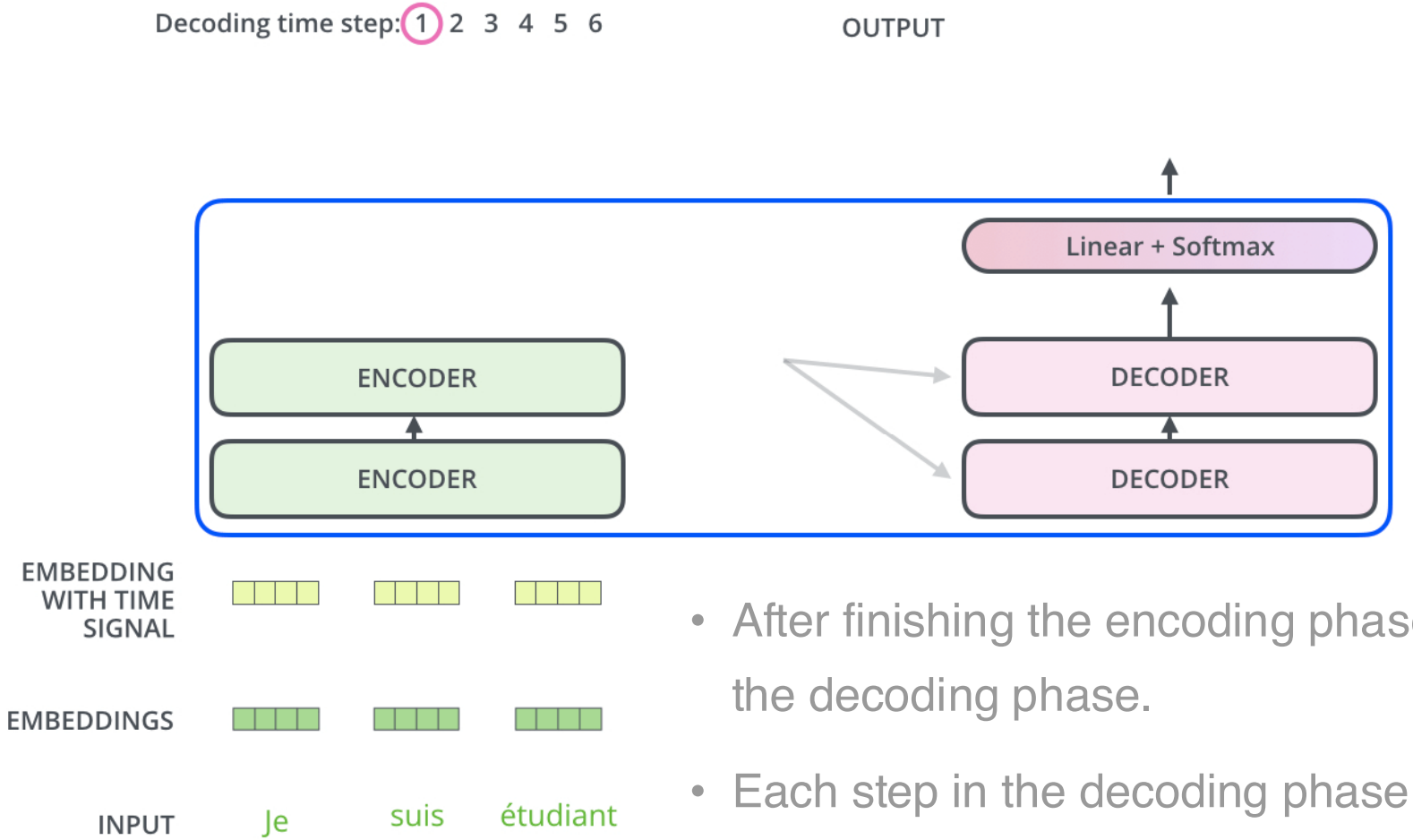
Add and Normalize



Encoder-decoder connection



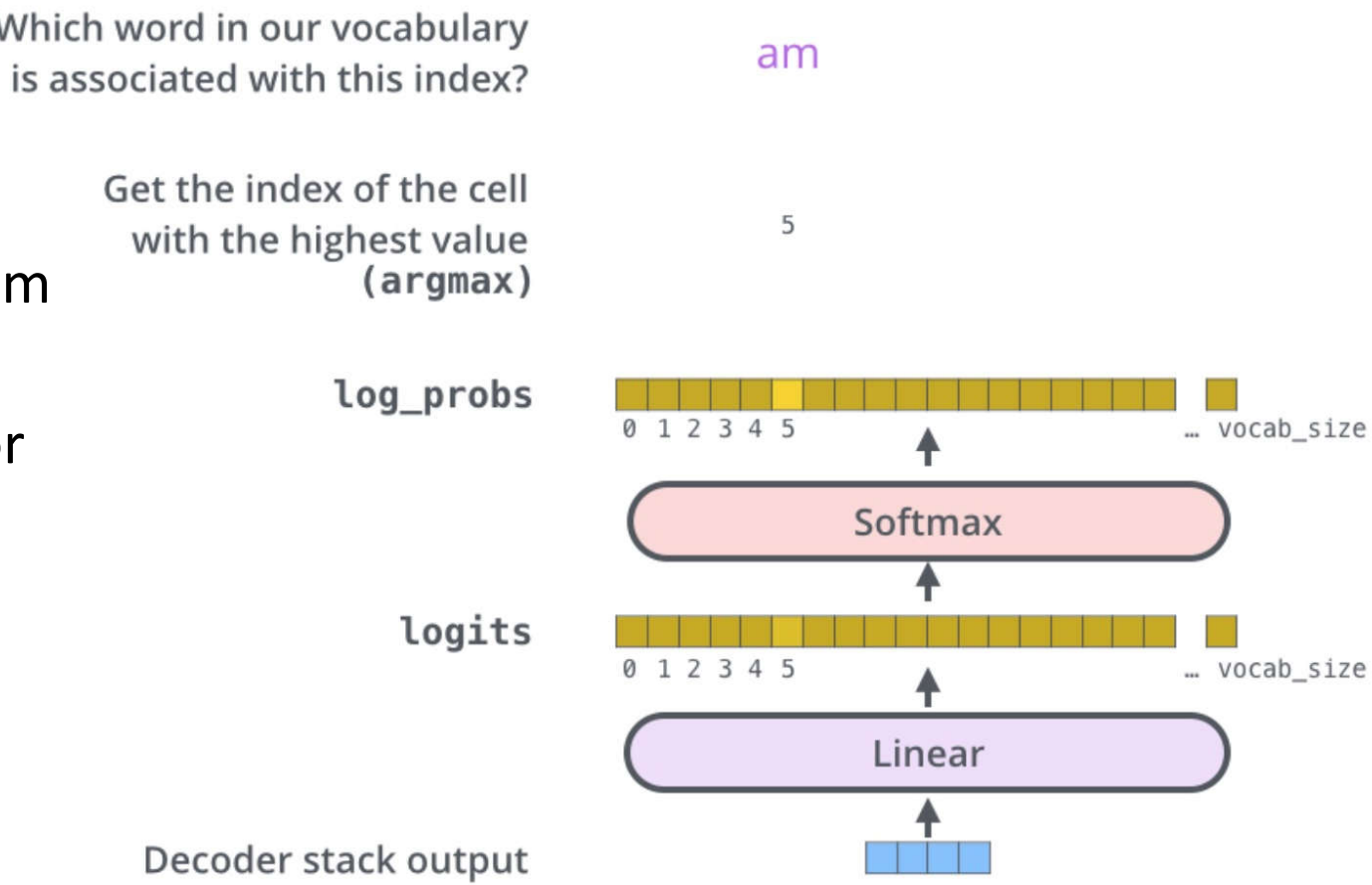
Decoder



- After finishing the encoding phase, we begin the decoding phase.
- Each step in the decoding phase outputs an element from the output sequence (the English translation sentence in this case)

Итоговое слово

- Let’s assume that our model knows 10,000 unique English words (our model’s “output vocabulary”) that it’s learned from its training dataset.
- This would make the logits vector 10,000 cells wide – each cell corresponding to the score of a unique word.
- That is how we interpret the output of the model followed by the Linear layer.



Model output