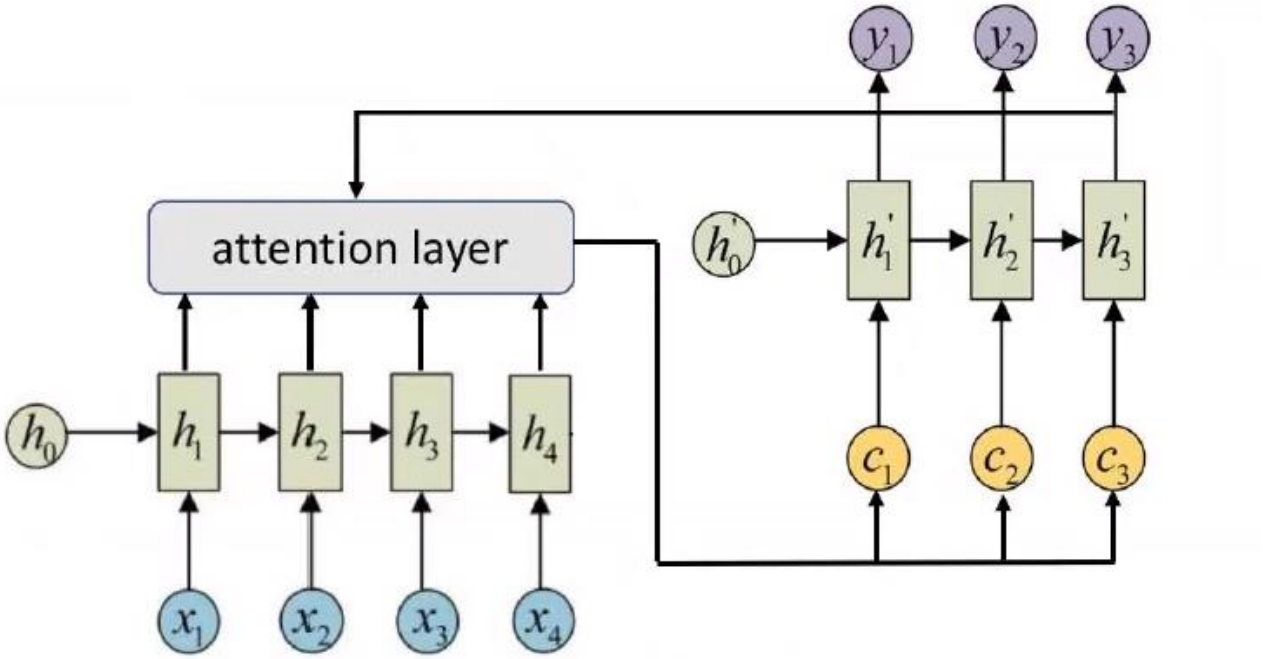
Трансформеры, модель BERT: архитектура и применение

ДОКЛАД ПОДГОТОВИЛ
АНДРЕЙ ЛЕБЕДЕВ

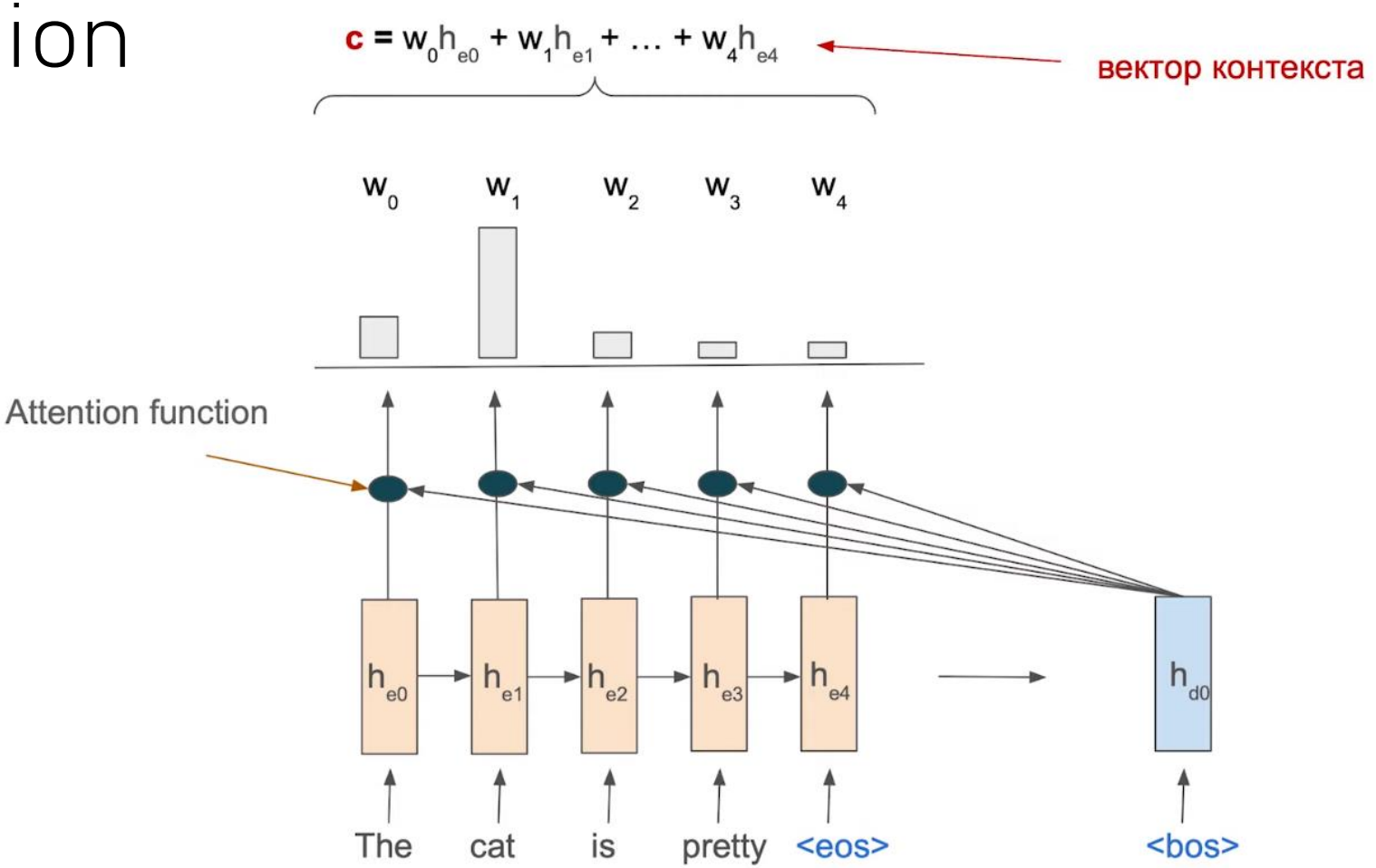
Содержание

Появление механизма внимания в RNN

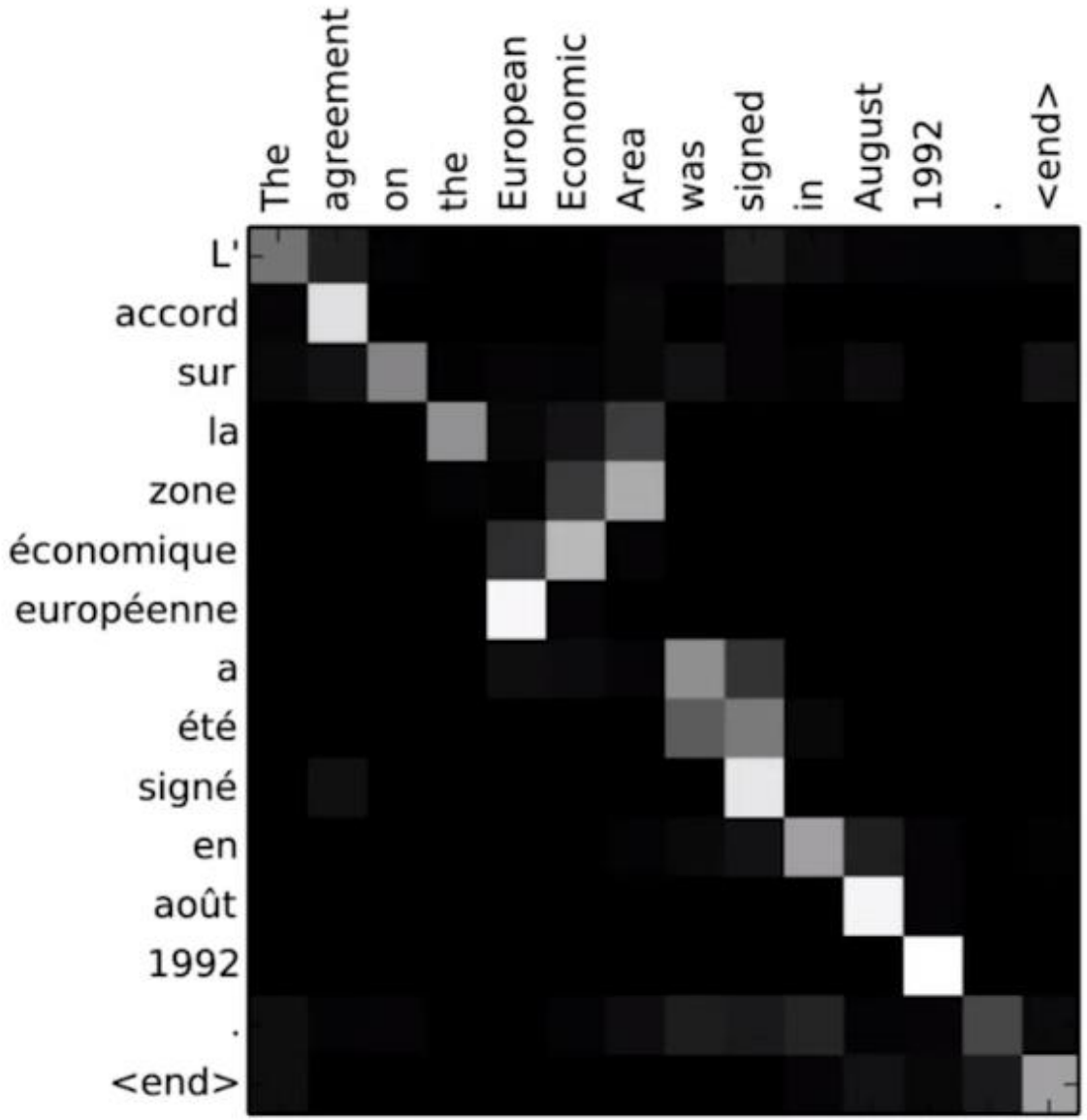
- The animal didn't cross the street because it was too tired.
- The animal didn't cross the street because it was too wide.



Attention

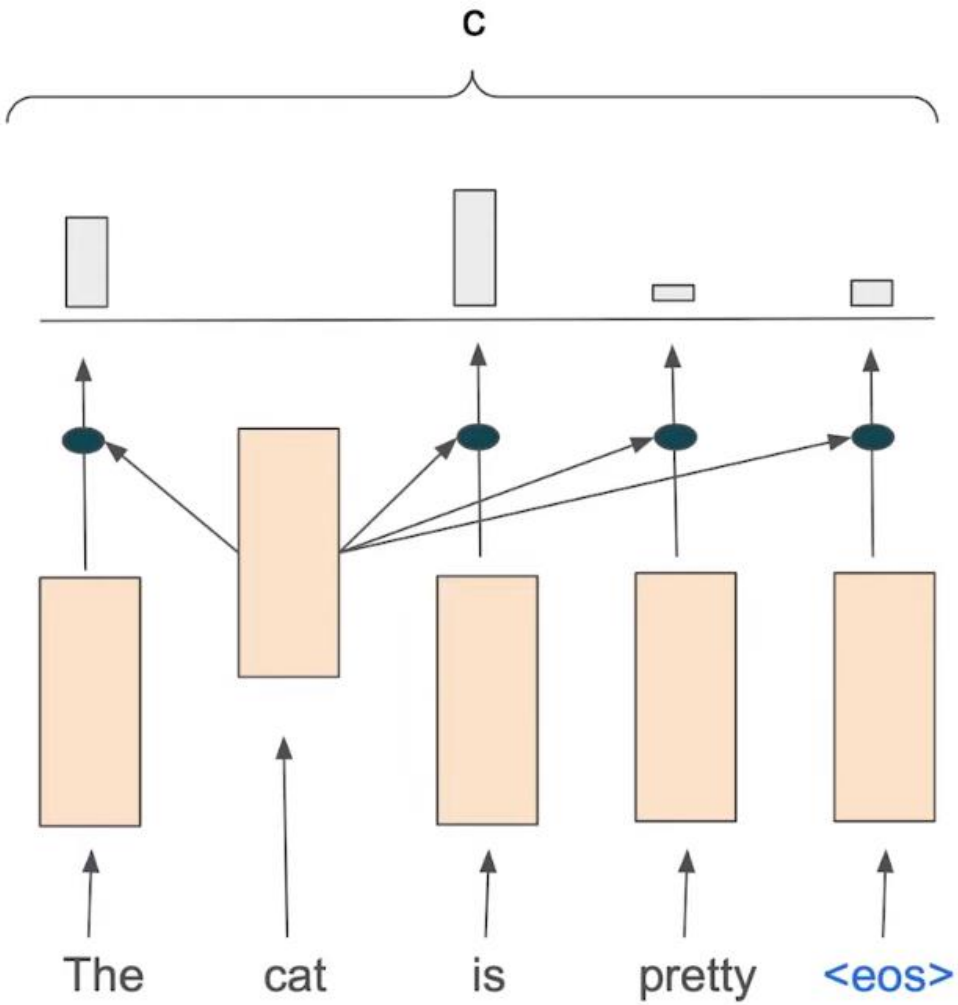

c = w0he0 + w1he1 + … + w4he4
вектор контекста
w0
w1
w2
w3
w4
Attention function
he0
he1
he2
he3
he4
hd0
The
cat
is
pretty
<eos>
<bos>

Пример работы attention



Self-attention

- A man in a nice cowboy hat and a parrot on his shoulder has entered the bar.
- Человек в красивой ковбойской шляпе и попугаем на плече … [зашел? зашла? зашло?]



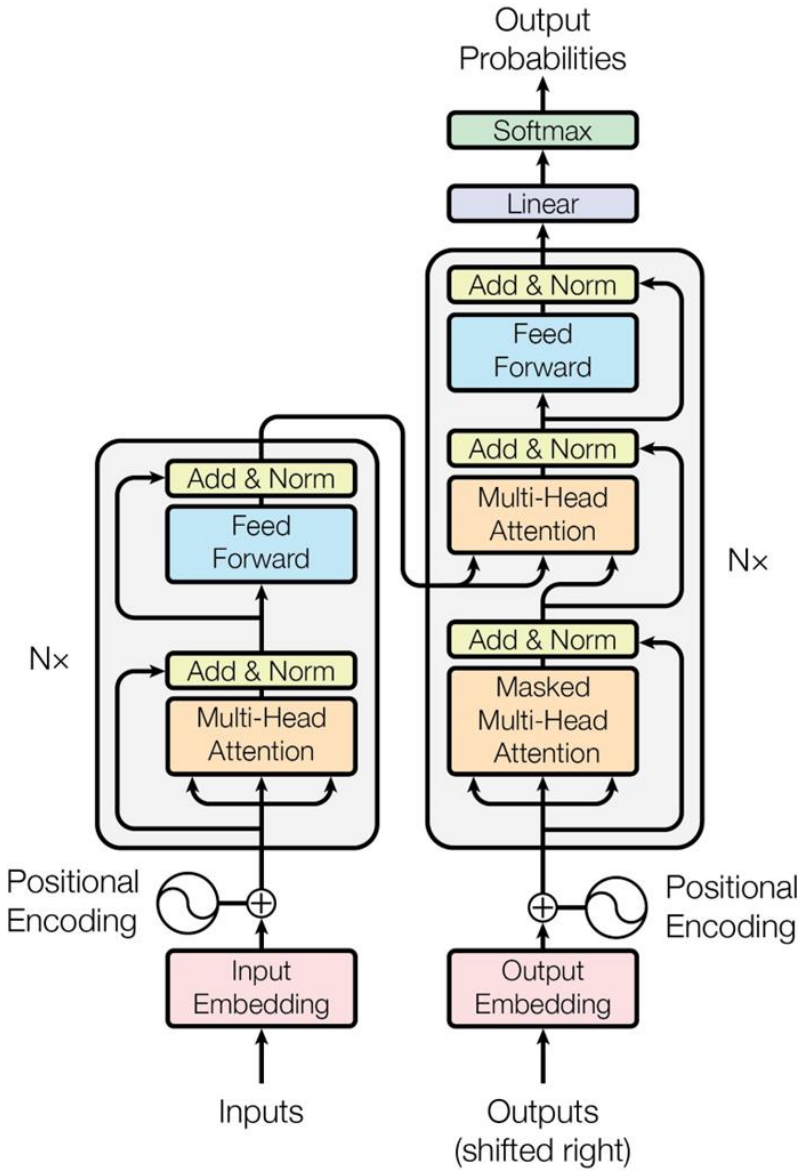
Multi-head attention

- Если в модели несколько механизмов внимания и само-внимания, такой механизм называется multi-head attention. При этом каждый из них может начать выделять определенную информацию, например, согласование по падежу/лицу и т. д.

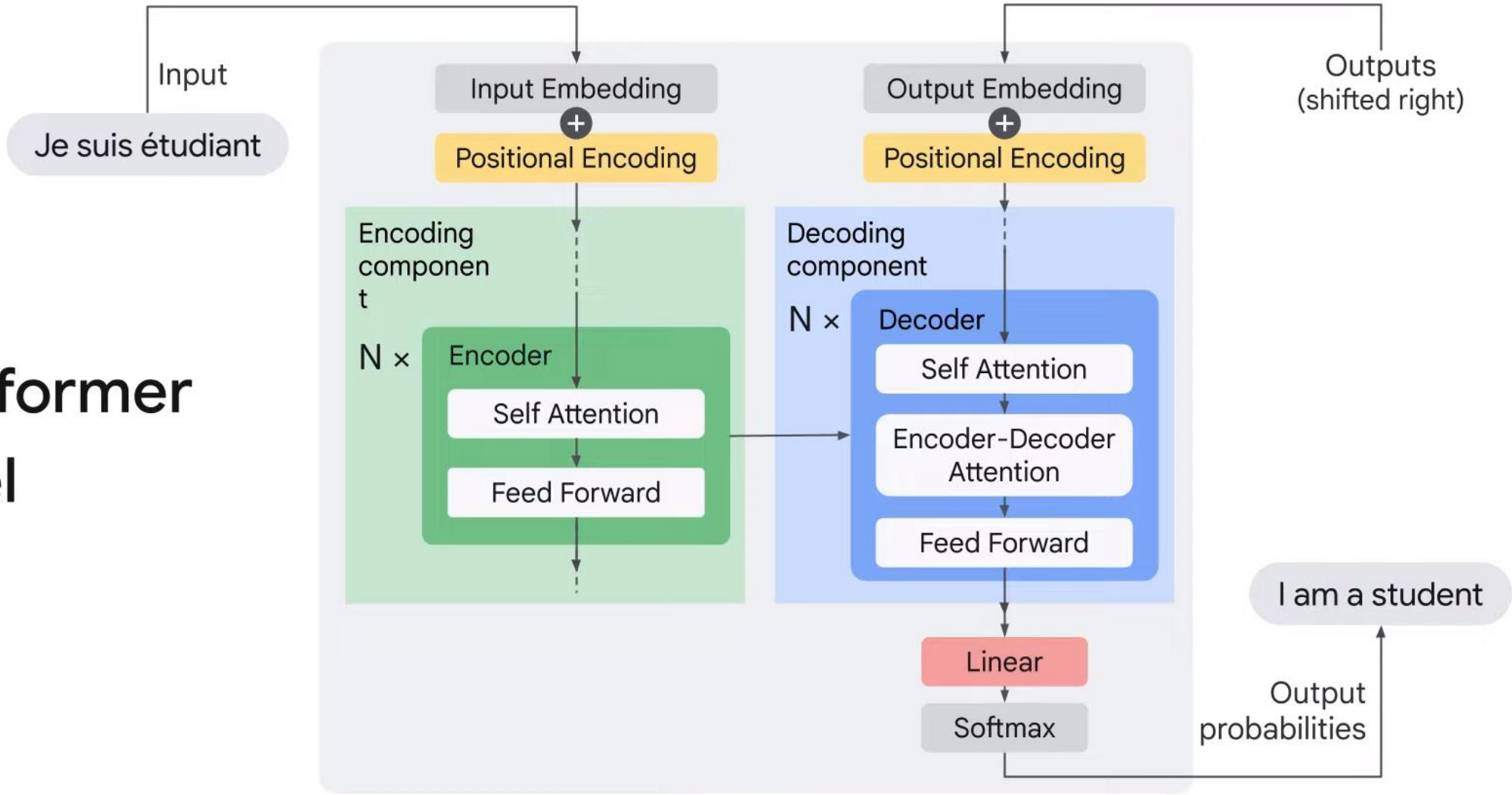
Трансформеры

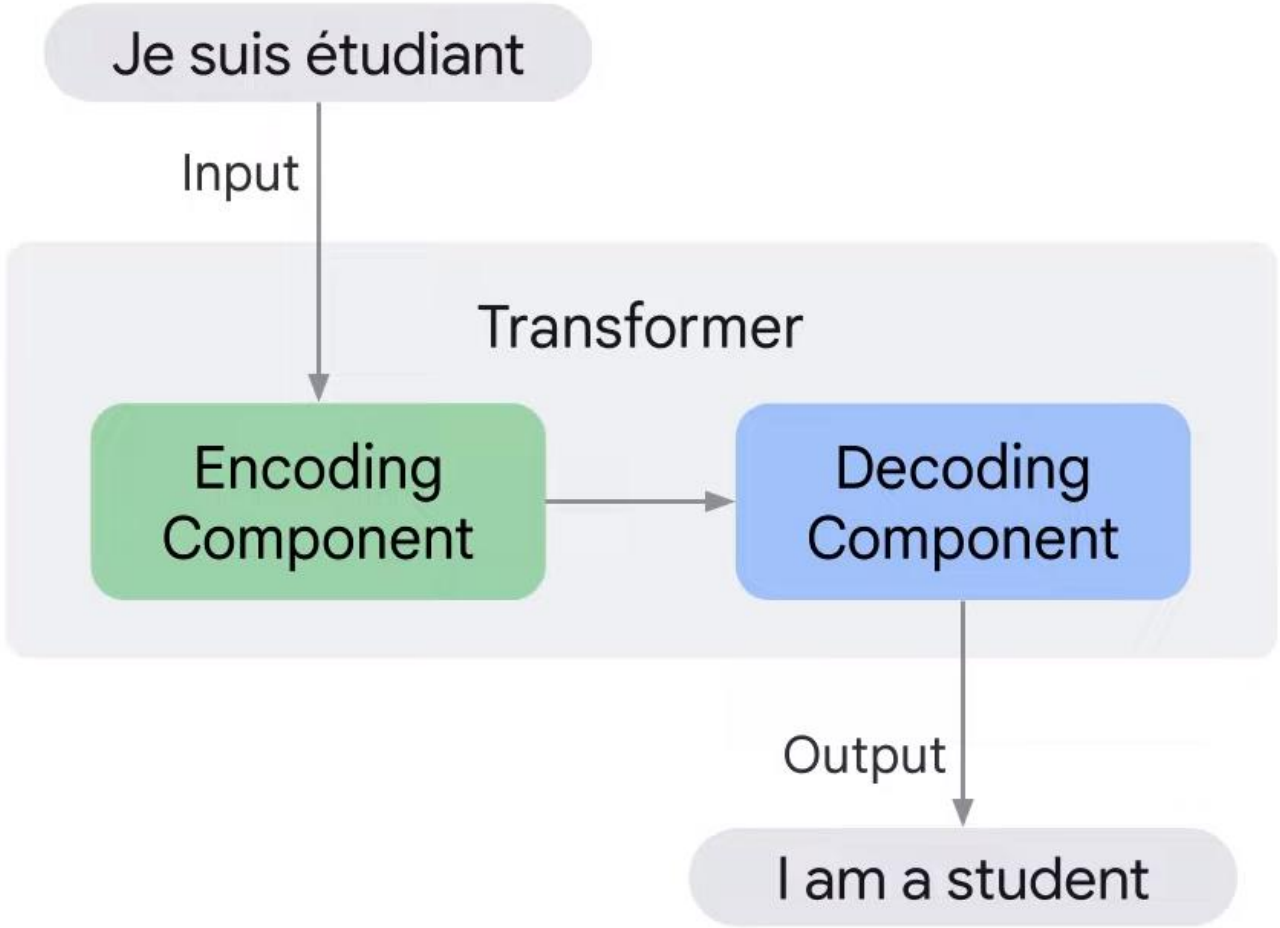
Трансформер

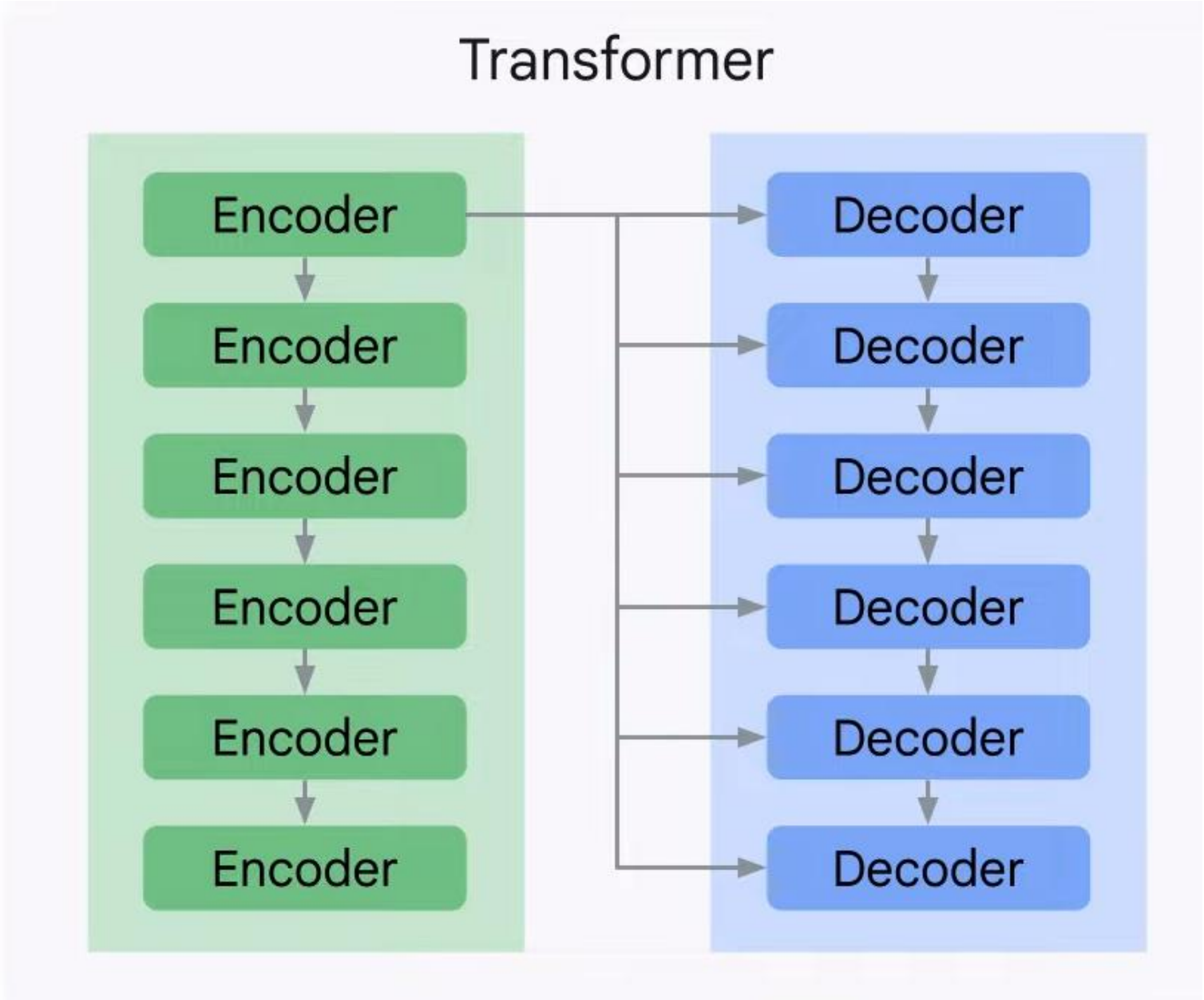
Трансформер – нейросетевая архитектура на основе внимания и полносвязных слоев.



Transformer model

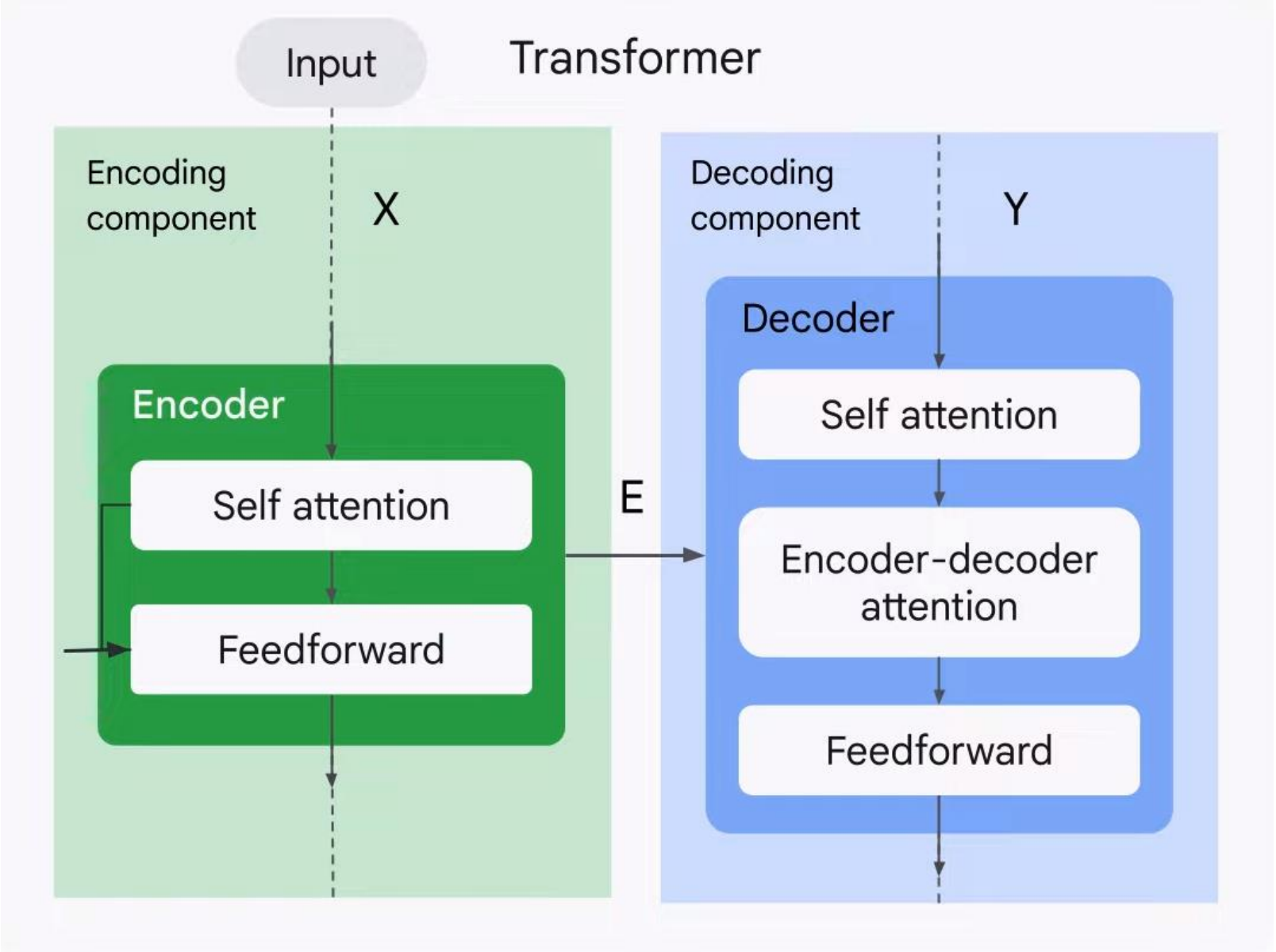


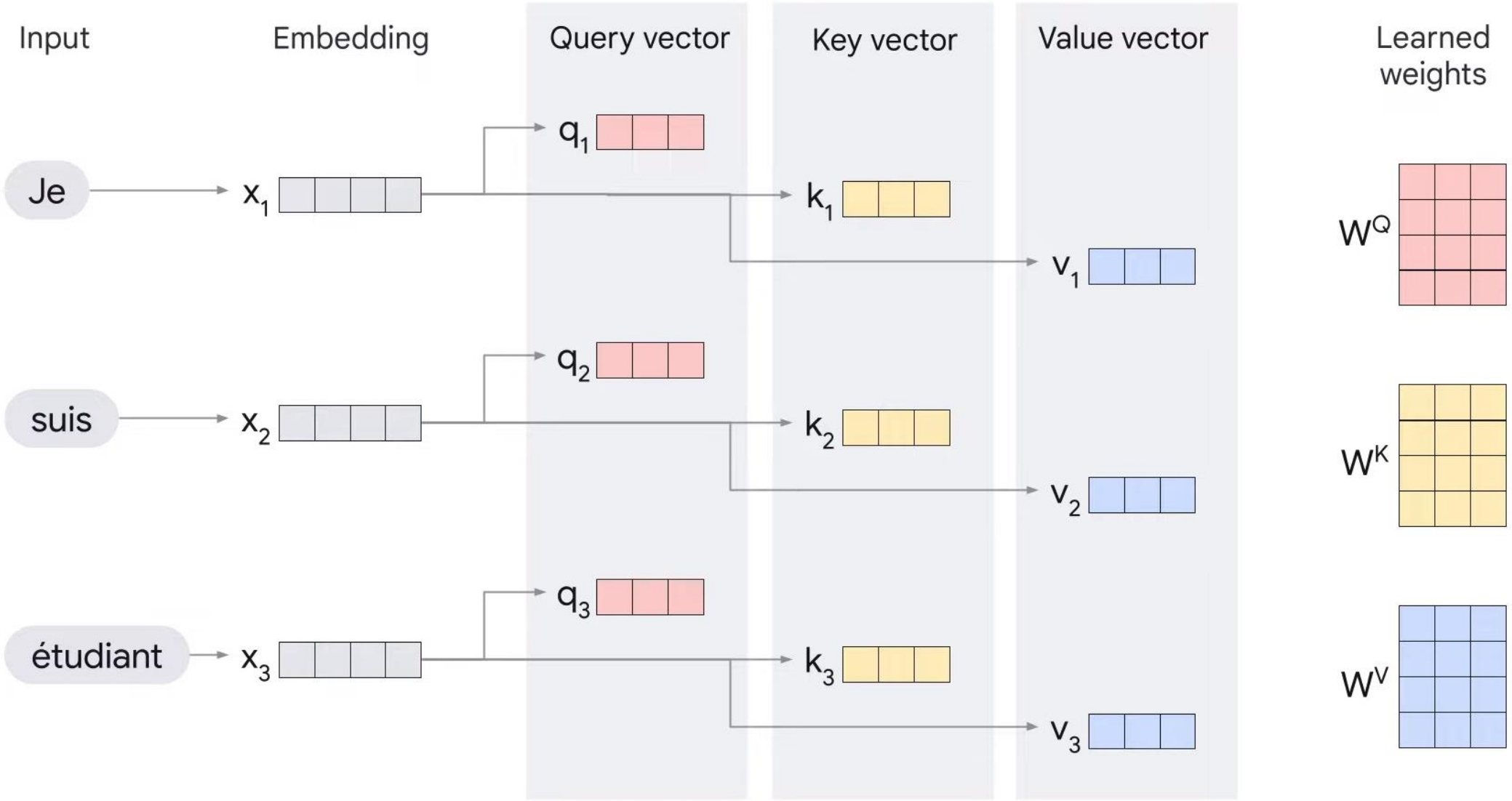

Je suis étudiant
Input
Transformer
Encoding Component
Decoding Component
Output
I am a student

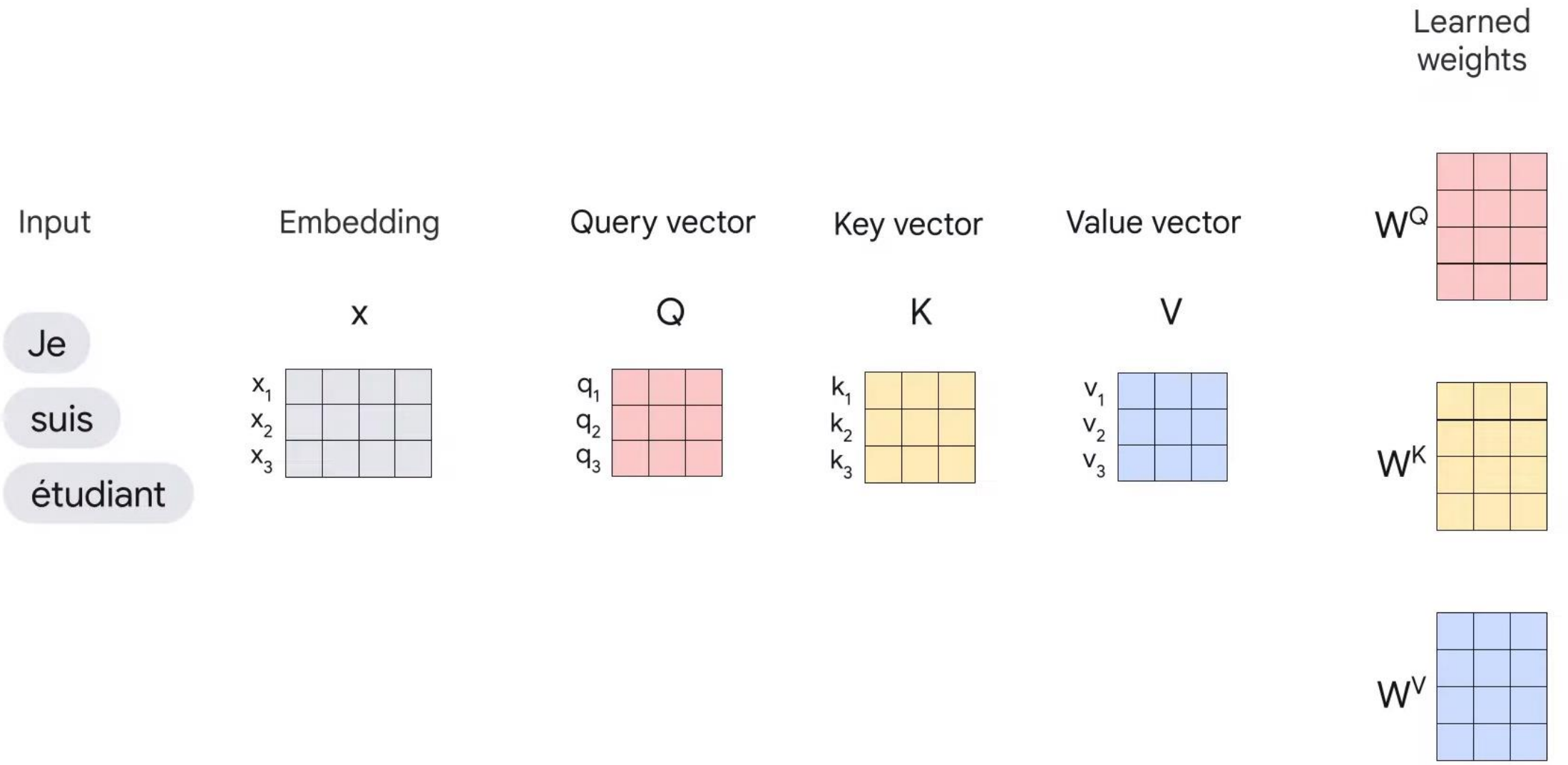

Transformer
Encoder
Encoder
Encoder
Encoder
Encoder
Encoder
Decoder
Decoder
Decoder
Decoder
Decoder
Decoder

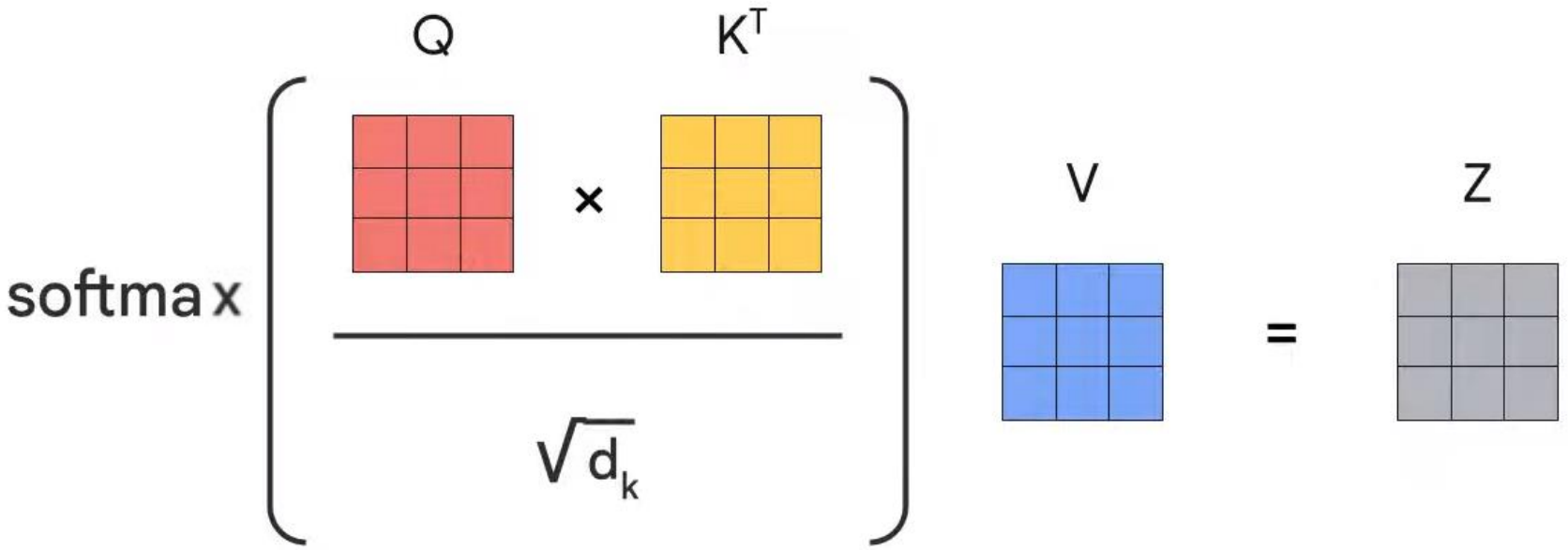
Feedforward – полносвязная сеть, позволяющая сделать обучаемое преобразование выходных данных со слоя внимания и подготовить их к следующему этапу кодирования:

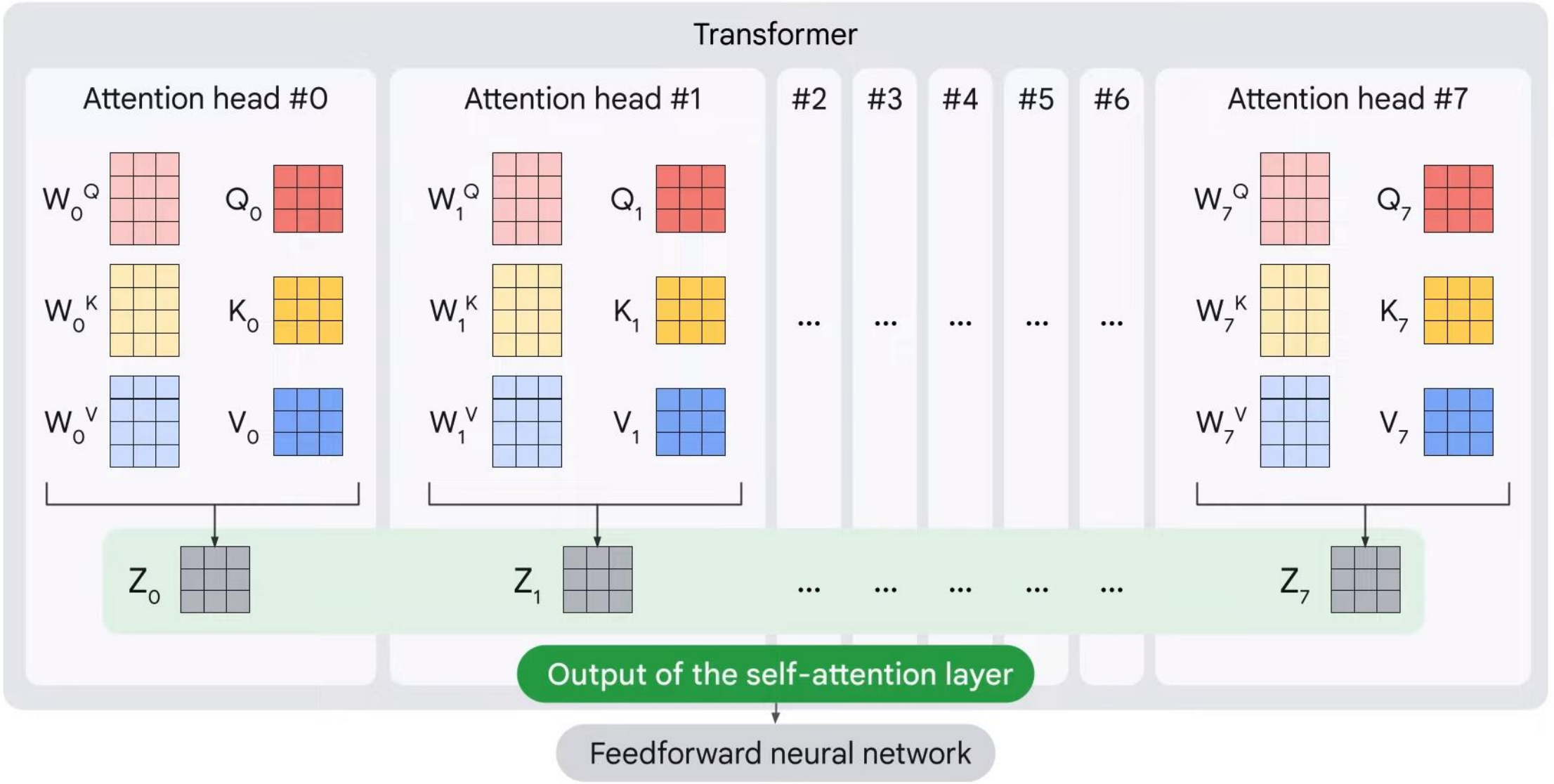
$$h_i''' = W_2 \mathrm{ReLU}(W_1 h_i'' + b_1) + b_2$$

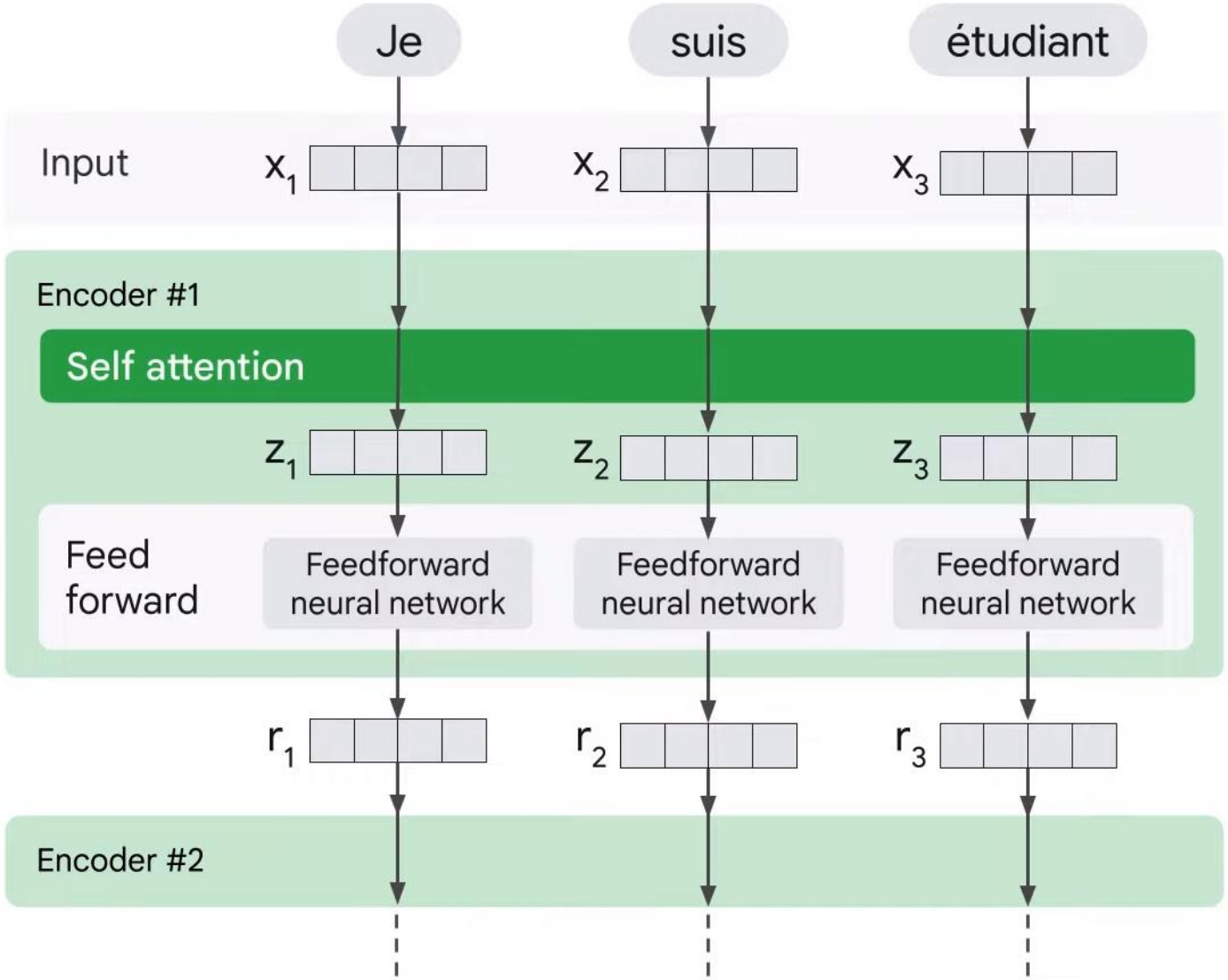



Input
Embedding
Query vector
Key vector
Value vector
Learned weights
Je
suis
étudiant
x_1
x_2
x_3
q_1
q_2
q_3
k_1
k_2
k_3
v_1
v_2
v_3
W^Q
W^K
W^V


Learned weights
Input
Embedding
Query vector
Key vector
Value vector
x
Q
K
V
Je
suis
étudiant
x_1
x_2
x_3
q_1
q_2
q_3
k_1
k_2
k_3
v_1
v_2
v_3
W^Q
W^K
W^V

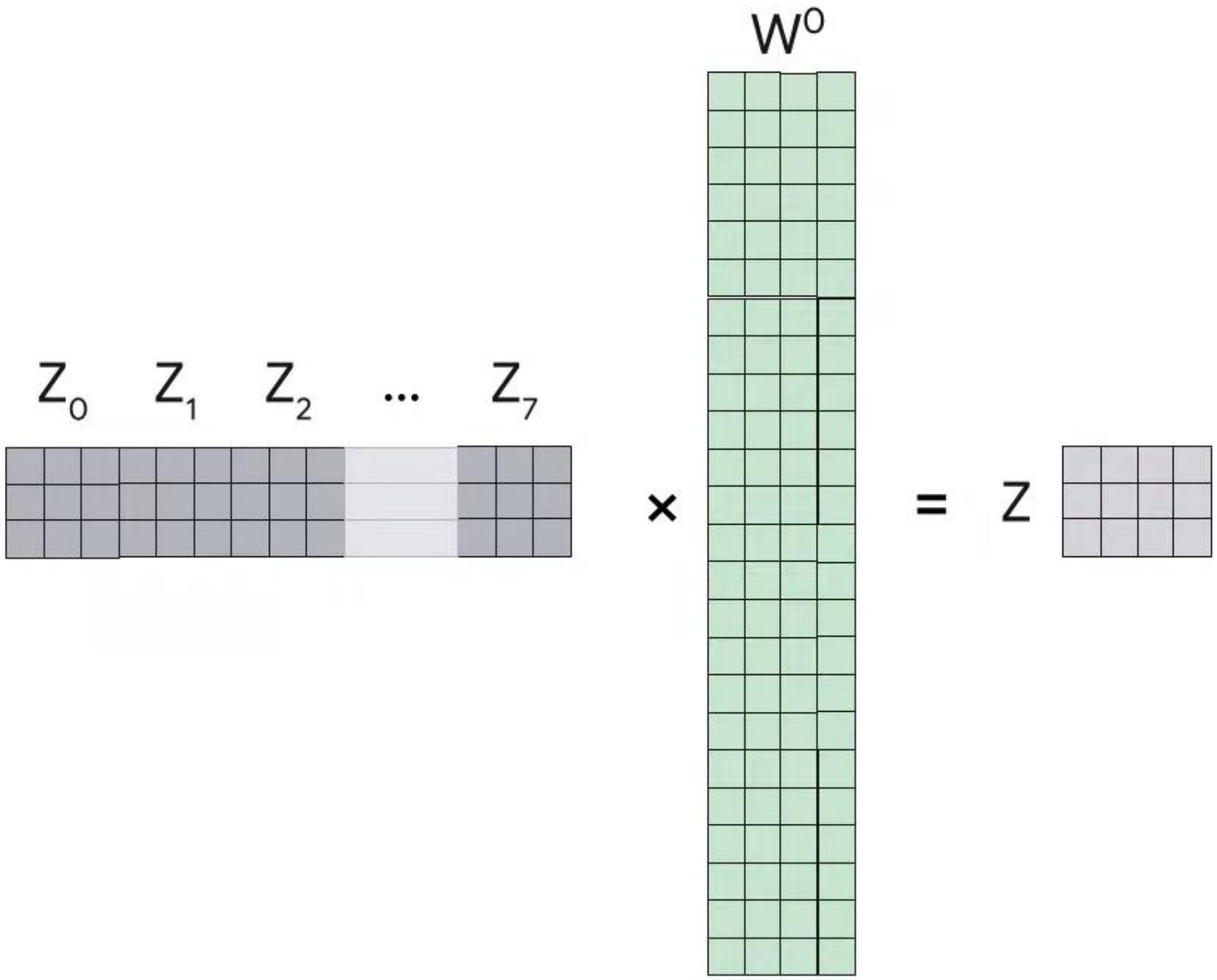

Q
K^T
softmax
√d_k
V
Z
=


Transformer
Attention head #0
Attention head #1
#2
#3
#4
#5
#6
Attention head #7
W_0^Q
Q_0
W_0^K
K_0
W_0^V
V_0
W_1^Q
Q_1
W_1^K
K_1
W_1^V
V_1
W_7^Q
Q_7
W_7^K
K_7
W_7^V
V_7
...
Z_0
Z_1
Z_7
Output of the self-attention layer
Feedforward neural network

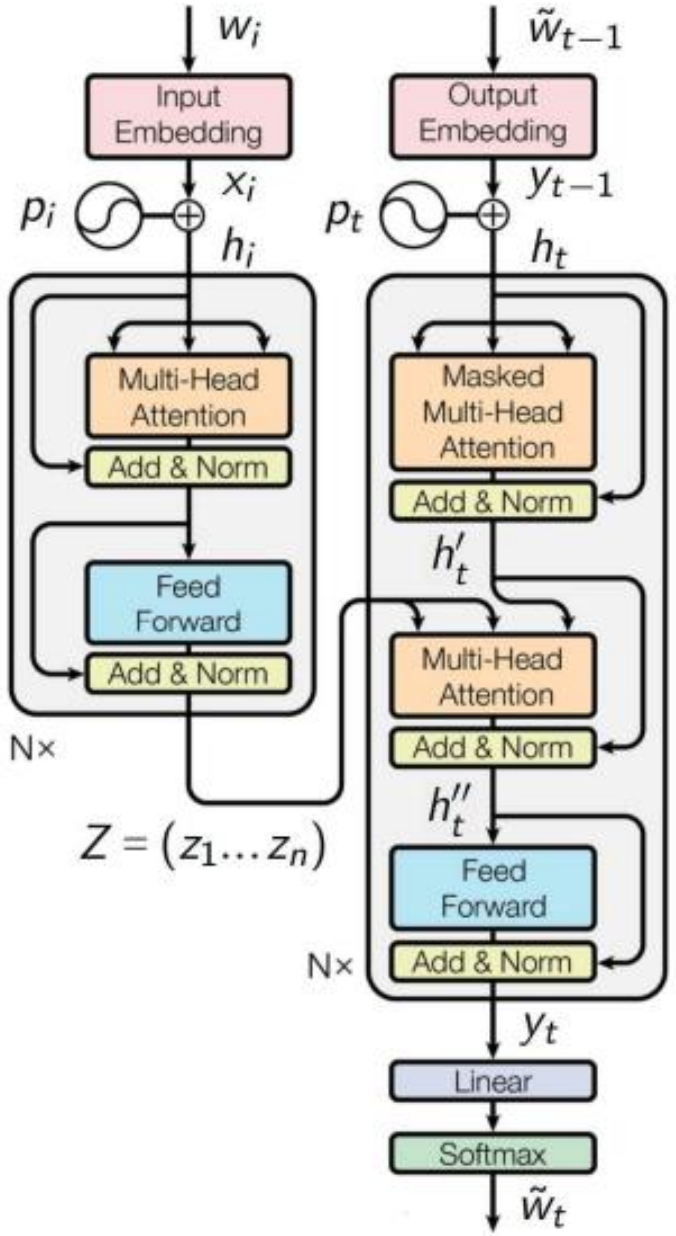

Je
suis
étudiant
Input
x_1
x_2
x_3
Encoder #1
Self attention
z_1
z_2
z_3
Feed forward
Feedforward neural network
Feedforward neural network
Feedforward neural network
r_1
r_2
r_3
Encoder #2

Процесс получения эмбеддингов z на выходе из энкодера

- Ввод предложения на естественном языке
- Векторизация каждого слова
- Выполнение multi-head attention и умножение векторизованных слов на соответствующие матрицы весов
- Вычисление функции внимания, используя полученные матрицы QKV
- Конкатенация матриц для получения выходной матрицы, которая имеет тот же размер, что и конечная матрица


W0
Z0 Z1 Z2 ... Z7
×
=
Z

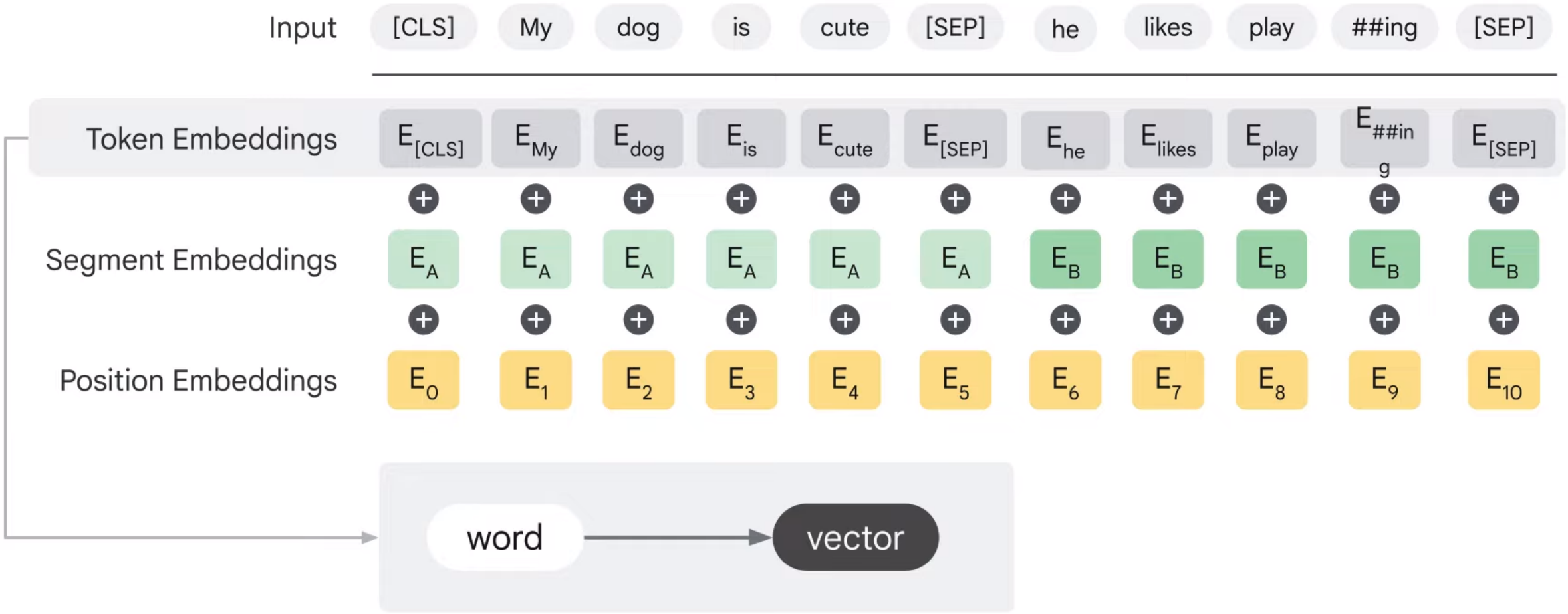
Резюмирование



BERT (Bidirectional Encoder Representations from Transformers)

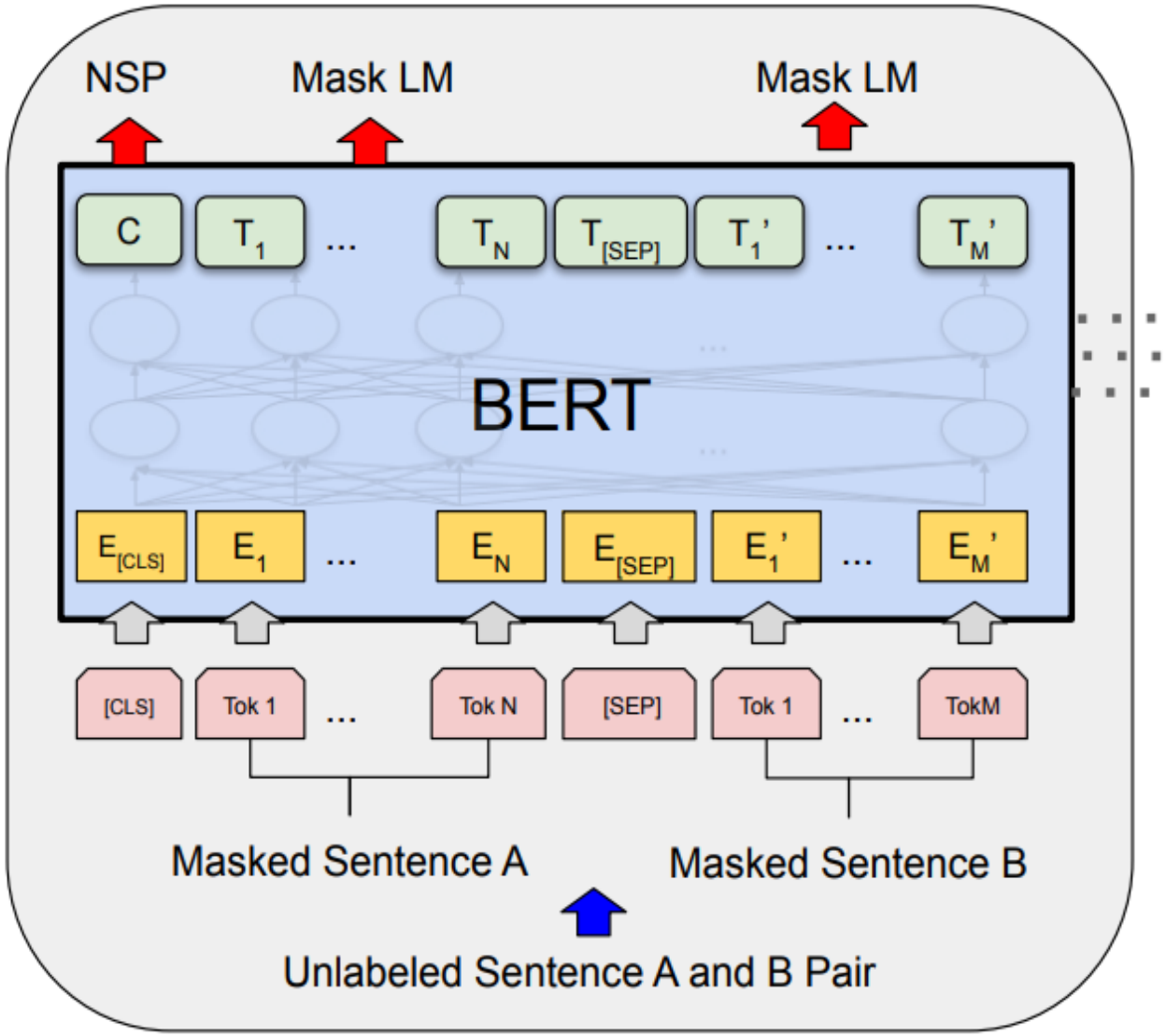
Общие положения

- Компания Google разработала языковую модель BERT в 2018 году. Она состоит из простого набора блоков-трансформеров, который был предварительно обучен на большом корпусе текстов общего характера.
- Отличается двунаправленностью внимания: это значит, что при обработке входной последовательности все токены могут использовать информацию друг о друге.


Input
[CLS] My dog is cute [SEP] he likes play ##ing [SEP]
Token Embeddings
E[CLS] EMy Edog Eis Ecute E[SEP] Ehe Elikes Eplay E##ing E[SEP]
Segment Embeddings
EA EA EA EA EA EA EB EB EB EB EB
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10
word
vector

Предобучение

- BERT обучается одновременно на двух задачах — предсказание следующего предложения и генерация пропущенного токена.
- На вход BERT подаются токенизированные пары предложений, в которых некоторые токены скрыты. Таким образом сеть обучается глубокому двунаправленному представлению языка, учится понимать контекст предложения.
- Задача же предсказания следующего предложения есть задача бинарной классификации — является ли второе предложение продолжением первого. Благодаря ей сеть можно обучить различать наличие связи между предложениями в тексте.


NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair

Какие задачи решает BERT?

1. Поисковый алгоритм Google
2. Классификация текстов
3. Ответы на вопросы
4. Генерация текста
5. Суммаризация текста

Спасибо за внимание!