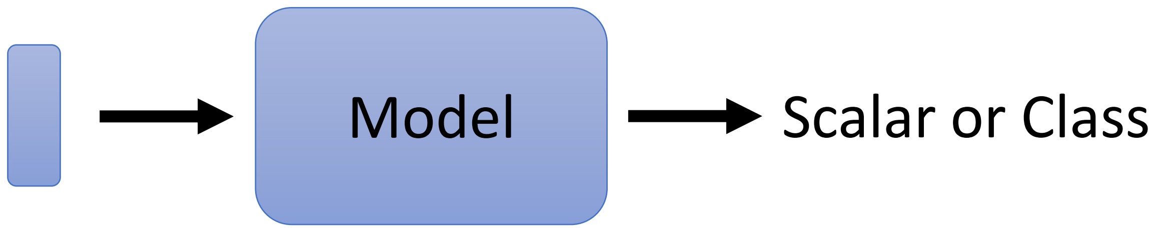
Self-attention

自注意力机制

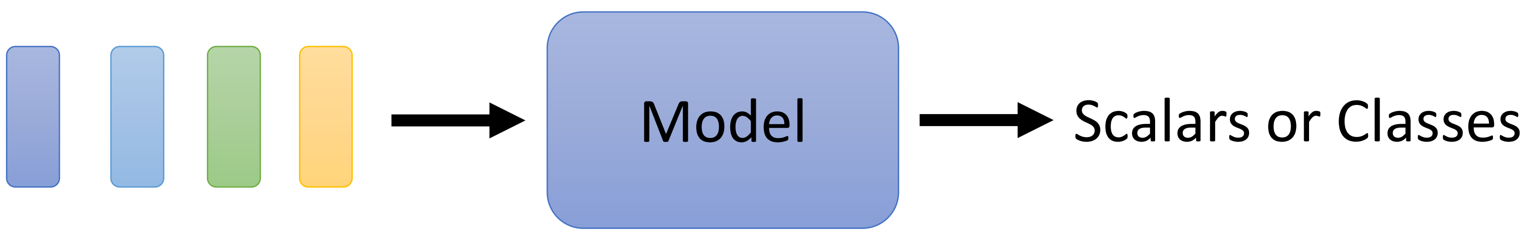
Hung-yi Lee
李宏毅

Sophisticated Input

- Input is **a vector** 之前的输入：一个向量

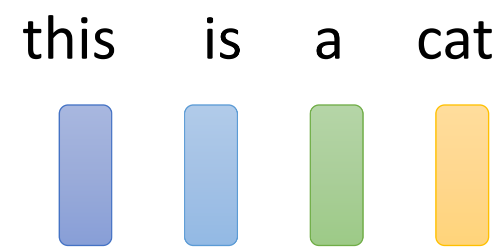


- Input is **a set of vectors** 现在我们开始考虑的输入：一排向量



(may change length)

Vector Set as Input



1 ***One-hot Encoding*** → 2 ***Word Embedding***

apple = [1 0 0 0 0 ……]

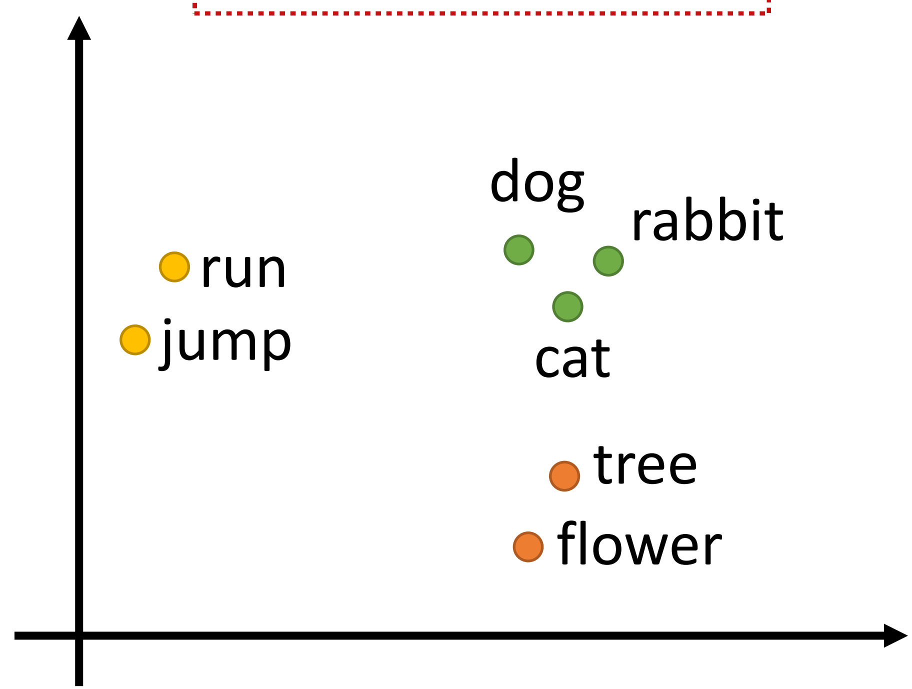
bag = [0 1 0 0 0 ……]

cat = [0 0 1 0 0 ……]

dog = [0 0 0 1 0 ……]

elephant = [0 0 0 0 1 ……]

1. length of vectors = # of words
2. 问题：假设word之间没有关系



To learn more: https://youtu.be/X7PH3NuYW0Q (in Mandarin)

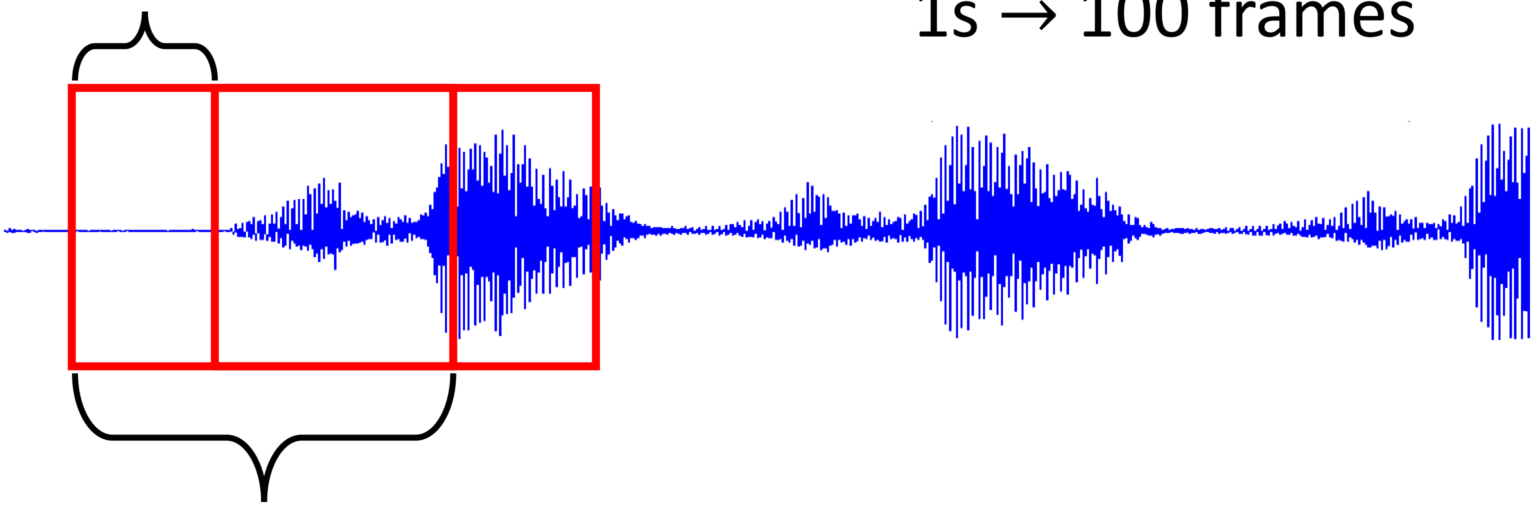
2017年ML课第14节 Word Embedding

Vector Set as Input

10ms （移动的大小）

1 min → 6,000 frames（数据量很客观）

1s → 100 frames



25ms （frame的大小）

frame

(window)

400 sample points (16KHz)

39-dim MFCC

80-dim filter bank output

https://medium.com/analytics-vidhya/social-network-analytics-f082f4e21b16

Vector Set as Input

- Graph is also a set of vectors (consider each **node** as **a vector**)

node：a vector
edge：有edge的两个nodes会计算关联程度α

Each profile is a vector

http://www.twword.com/wiki/%E5%88%86%E5%AD%90

Vector Set as Input

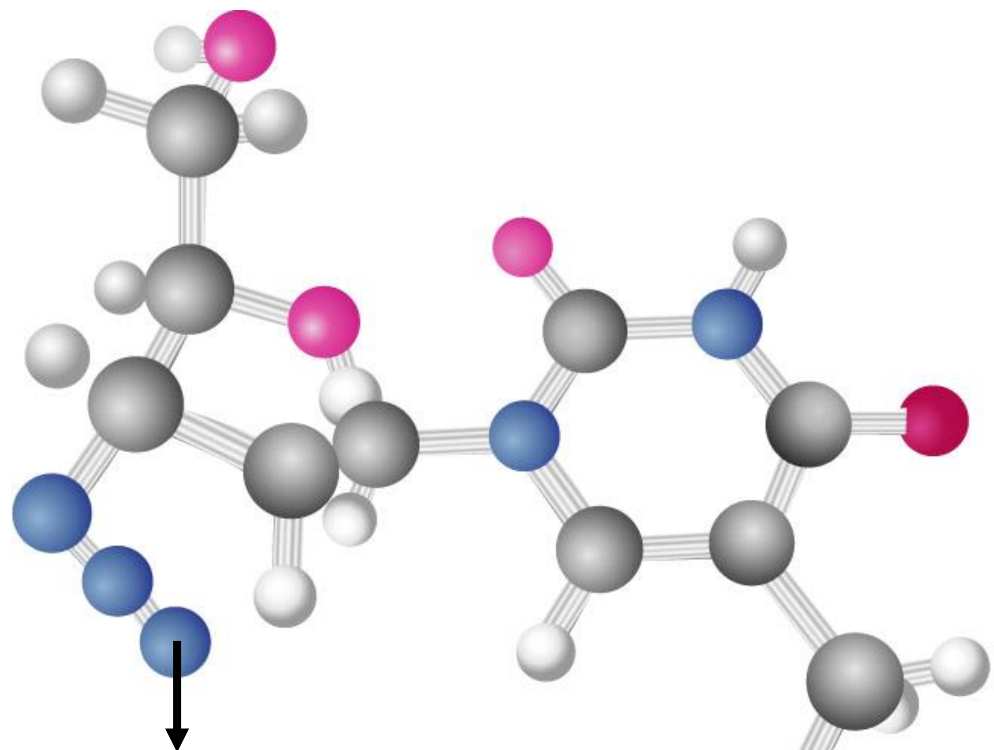
- Graph is also a set of vectors (consider each **node** as **a vector**)

H = [1 0 0 0 0 ……]

C = [0 1 0 0 0 ……]

O = [0 0 1 0 0 ……]

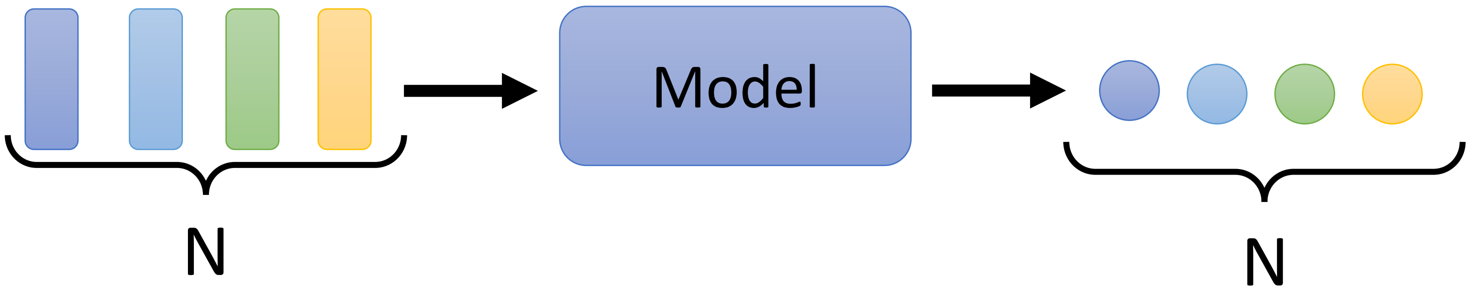
⋮



One-hot vector

What is the output? 有三种可能性。

1. • Each vector has a label.



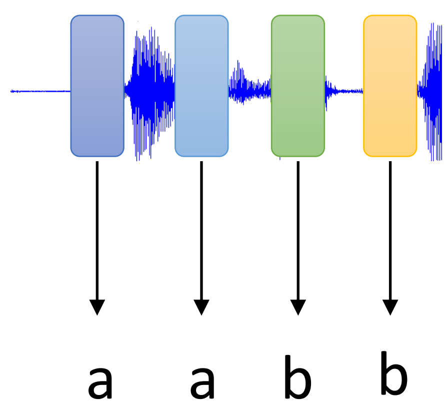
Example Applications

v.看见 n.锯子

I saw a saw

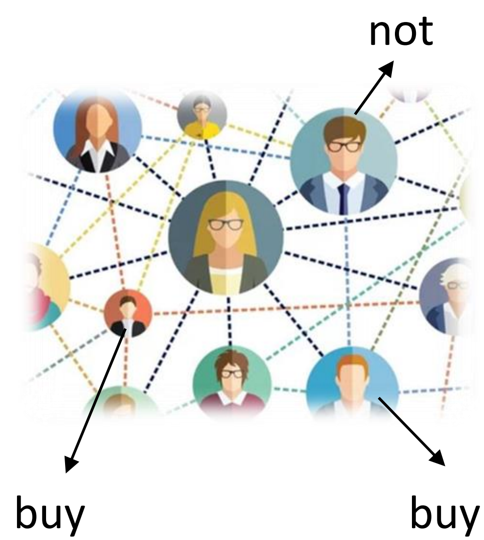
N V DET N

POS tagging



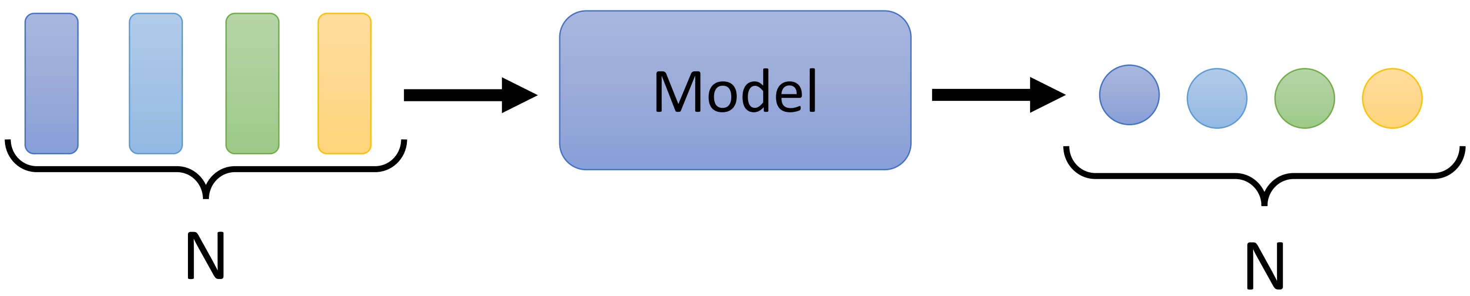
HW2

语音识别的简化版

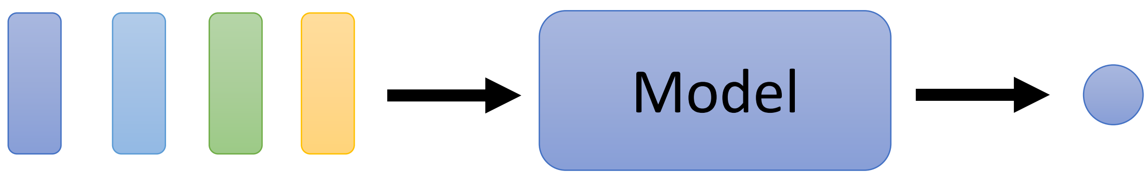


What is the output?

1. • Each vector has a label.



2. • The whole sequence has a label. Sequence-to-vector model



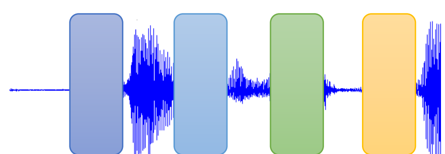
Example Applications

this is good

Sentiment analysis
情感分析

↓

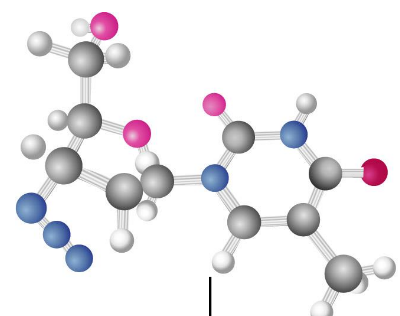
positive



↓

HW4

speaker 语者分析



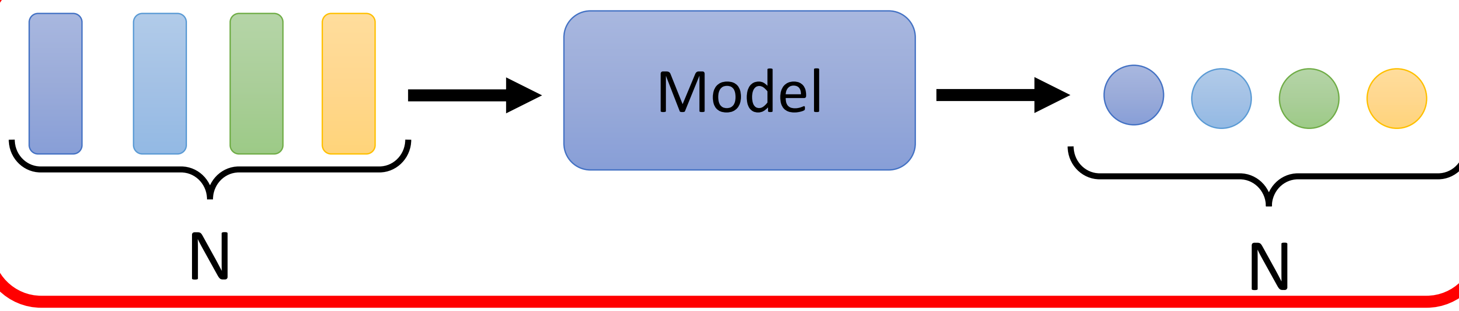
↓

亲水性

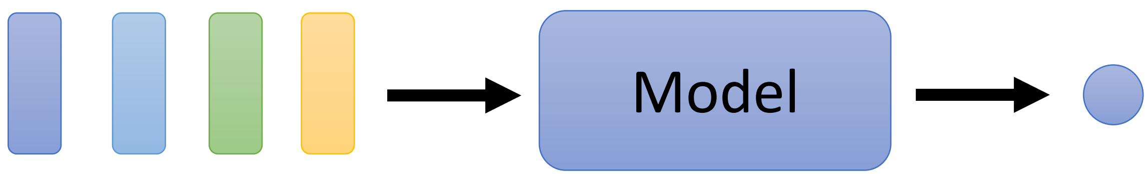
hydrophilicity

What is the output?

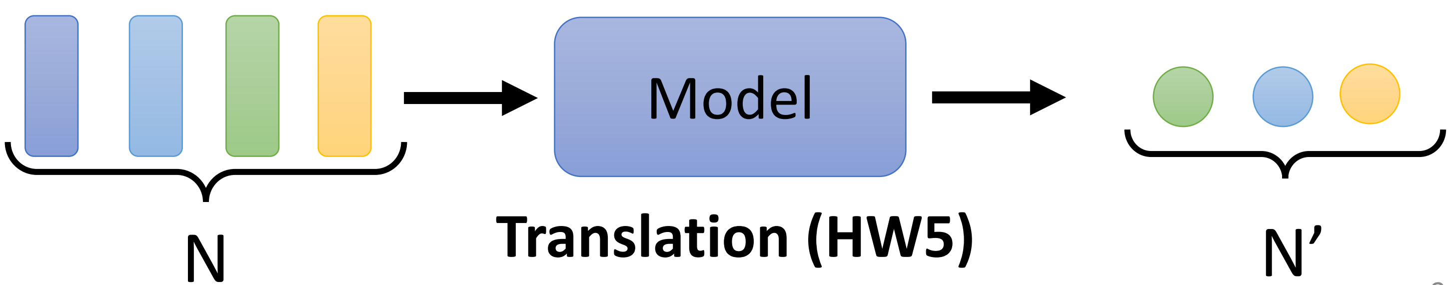
- 1 • Each vector has a label. focus of this lecture



- 2 • The whole sequence has a label.



- 3 • Model decides the number of labels itself. seq2seq



Sequence Labeling

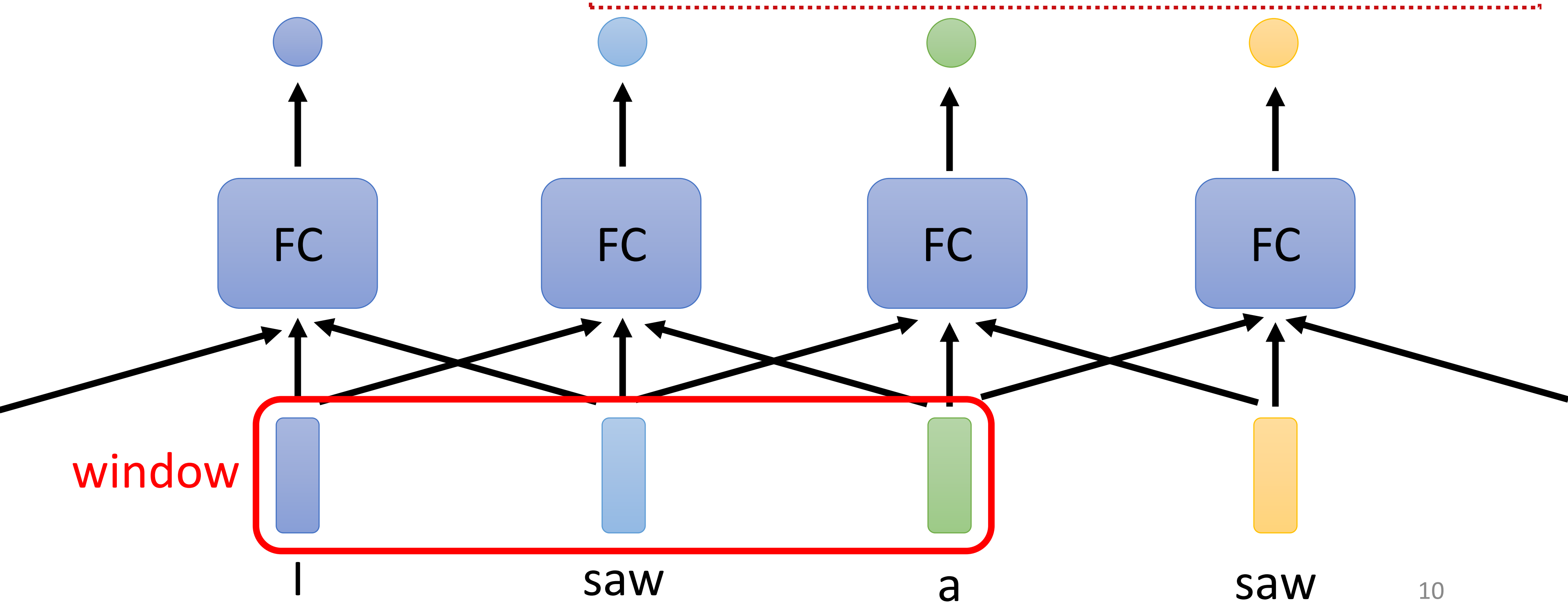
Is it possible to consider the context?

FC can consider the neighbor

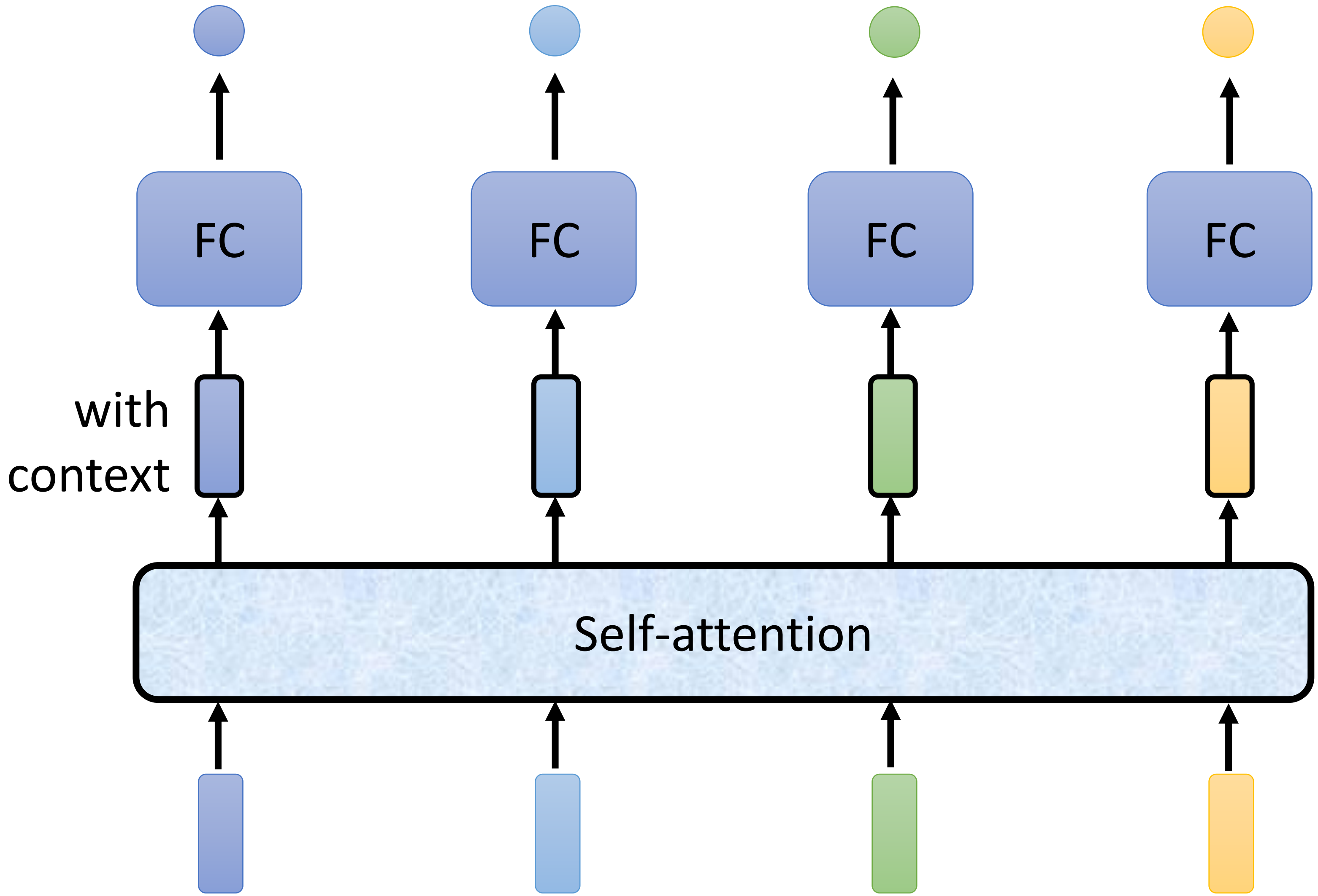
How to consider the whole sequence?

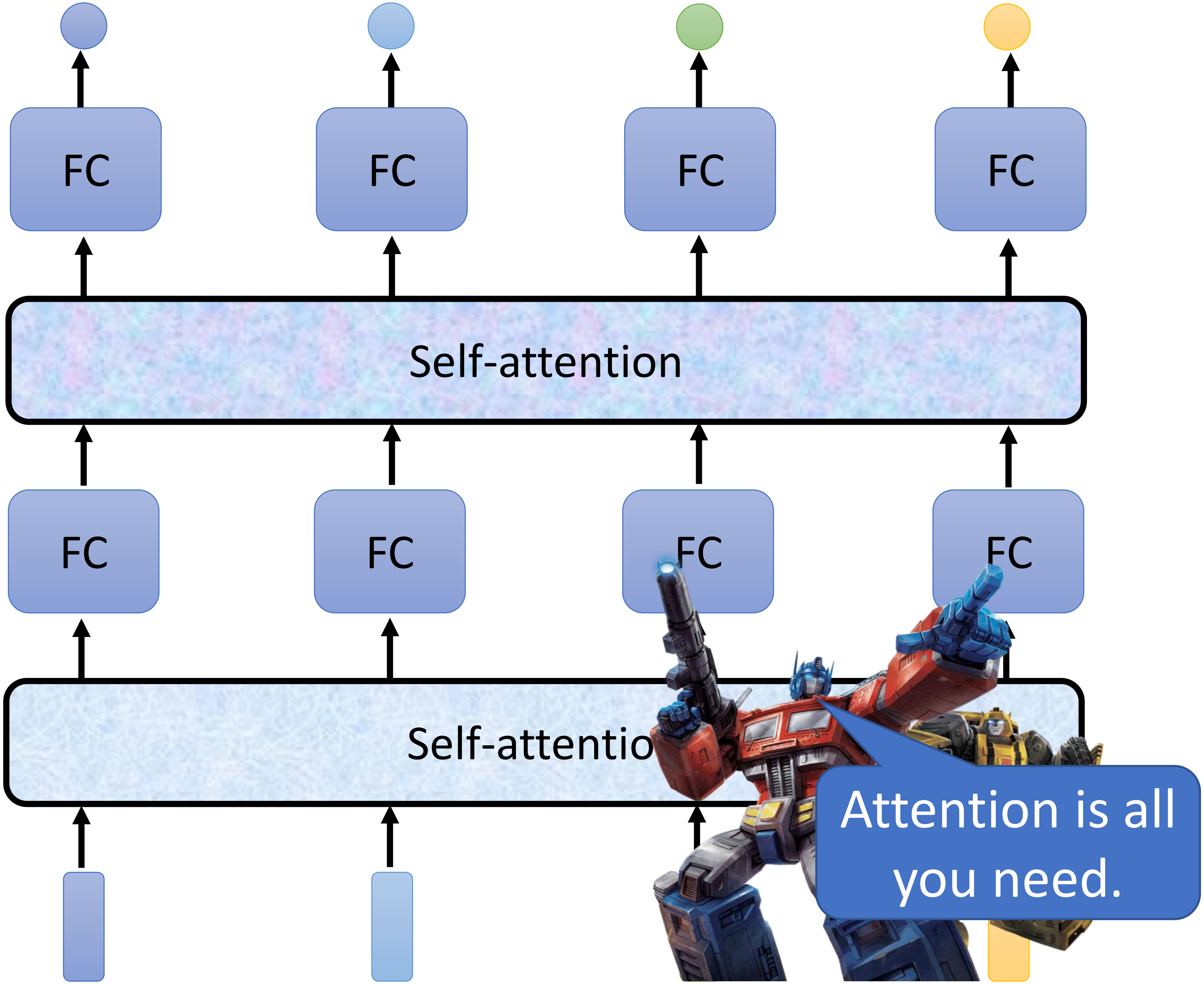
a window covers the whole sequence?

FC Fully-connected



Self-attention

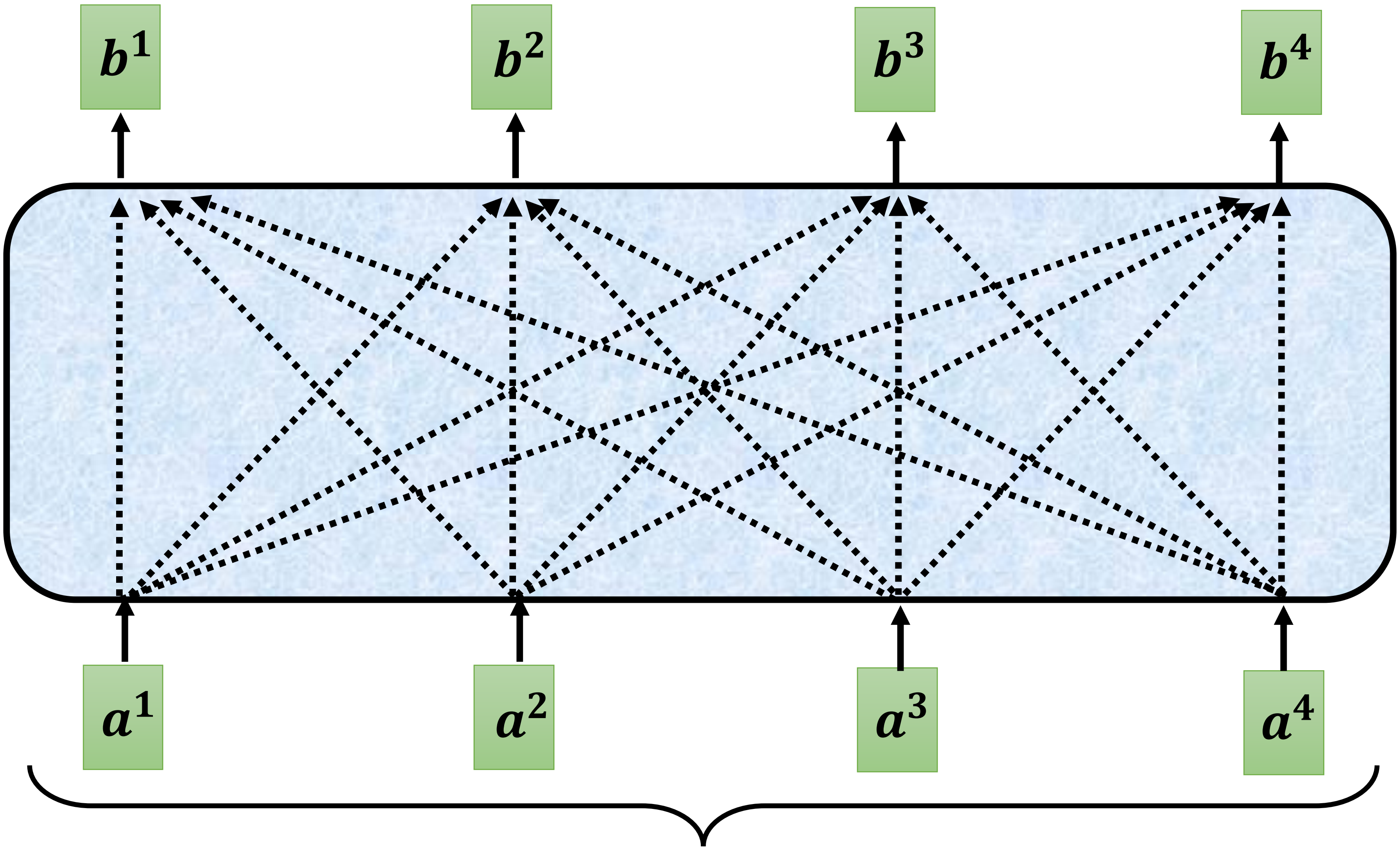




self-attention层可以叠加

https://arxiv.org/abs/1706.03762

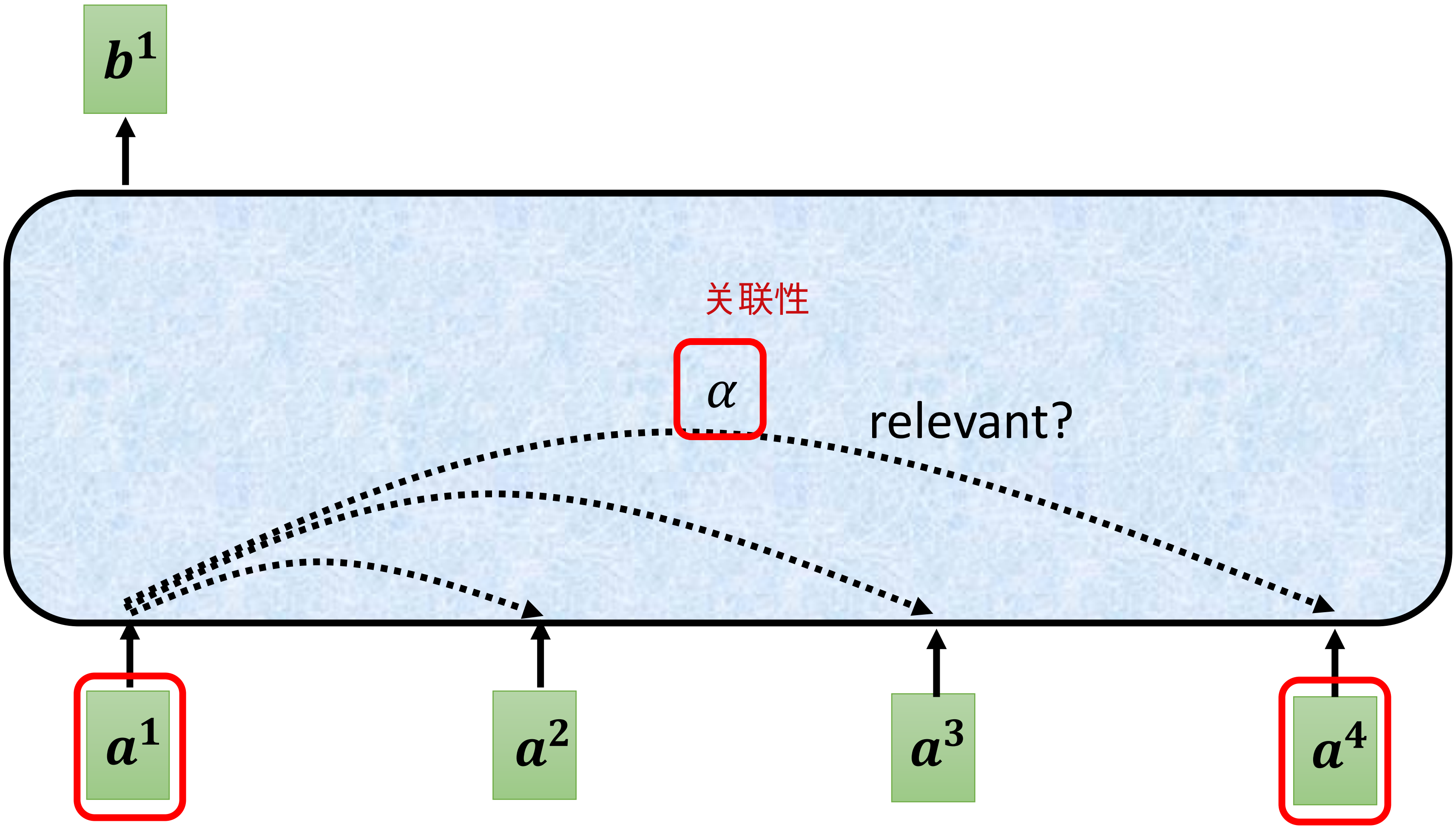
Self-attention



Can be either **input** or **a hidden layer**

Self-attention

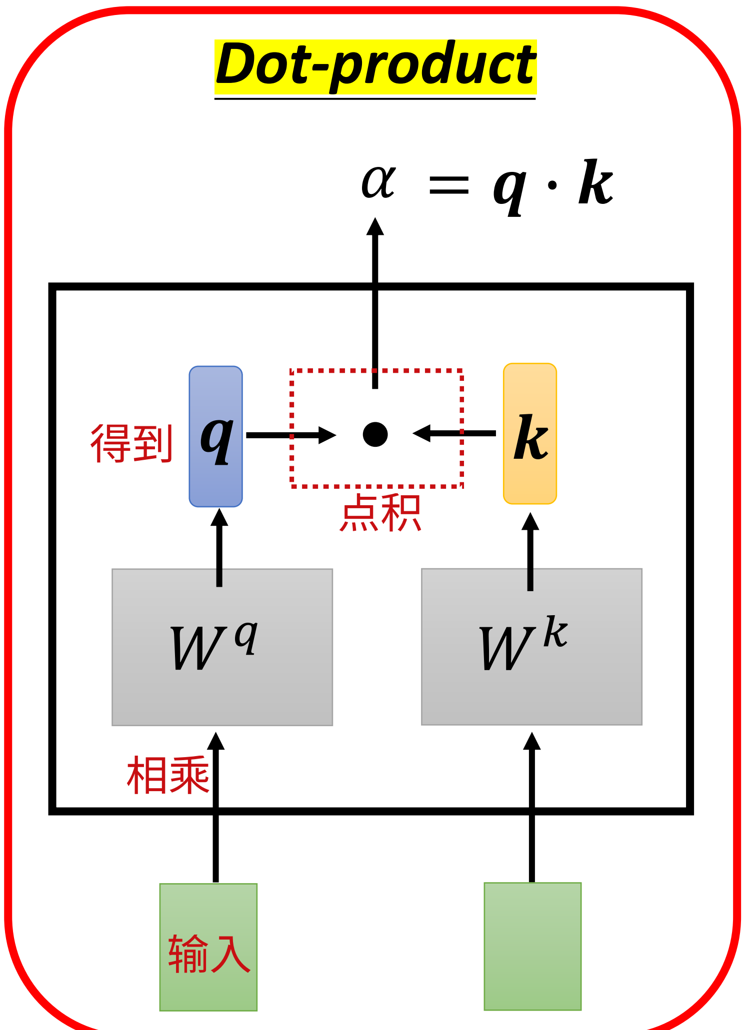
怎么产生b1呢？
第一步：根据找出a1和其他向量的关联程度α



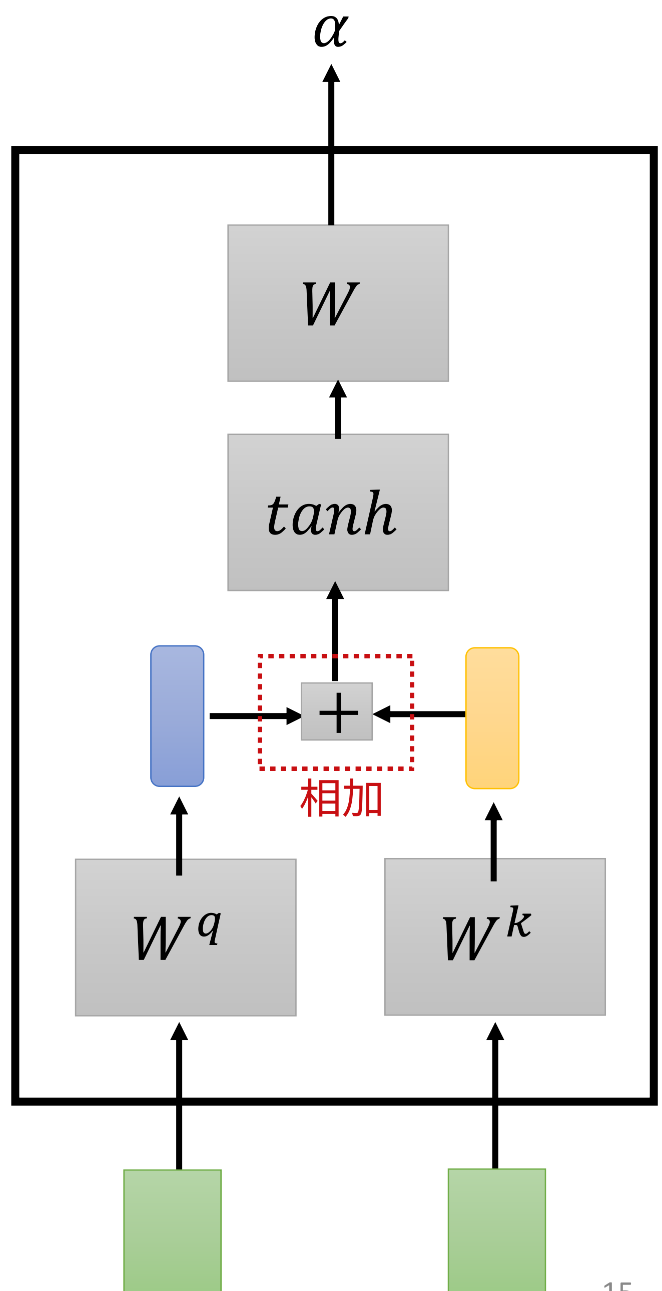
Find the relevant vectors in a sequence

如何计算α呢？两种常见的方法。
1. dot-product
2. additive

Self-attention



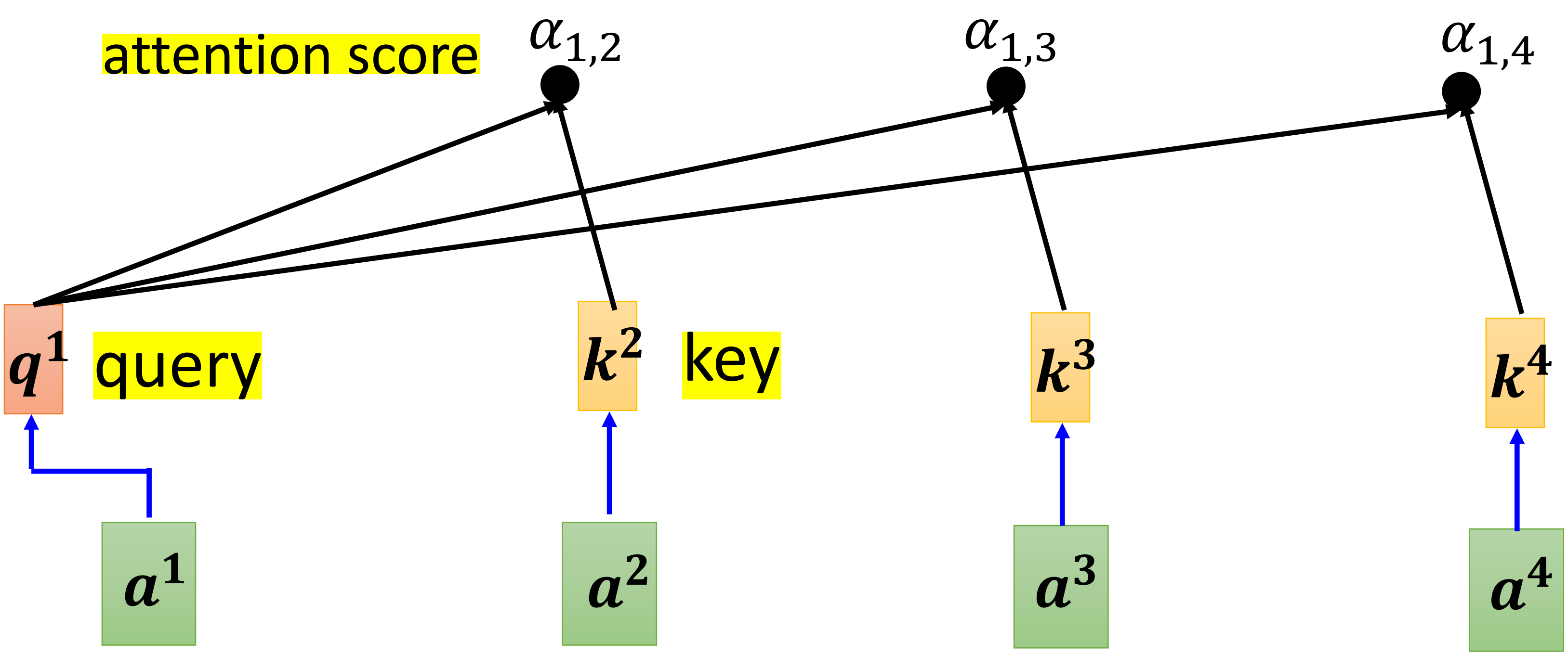
Additive



Self-attention

1. q：query
2. k：key
3. attention score

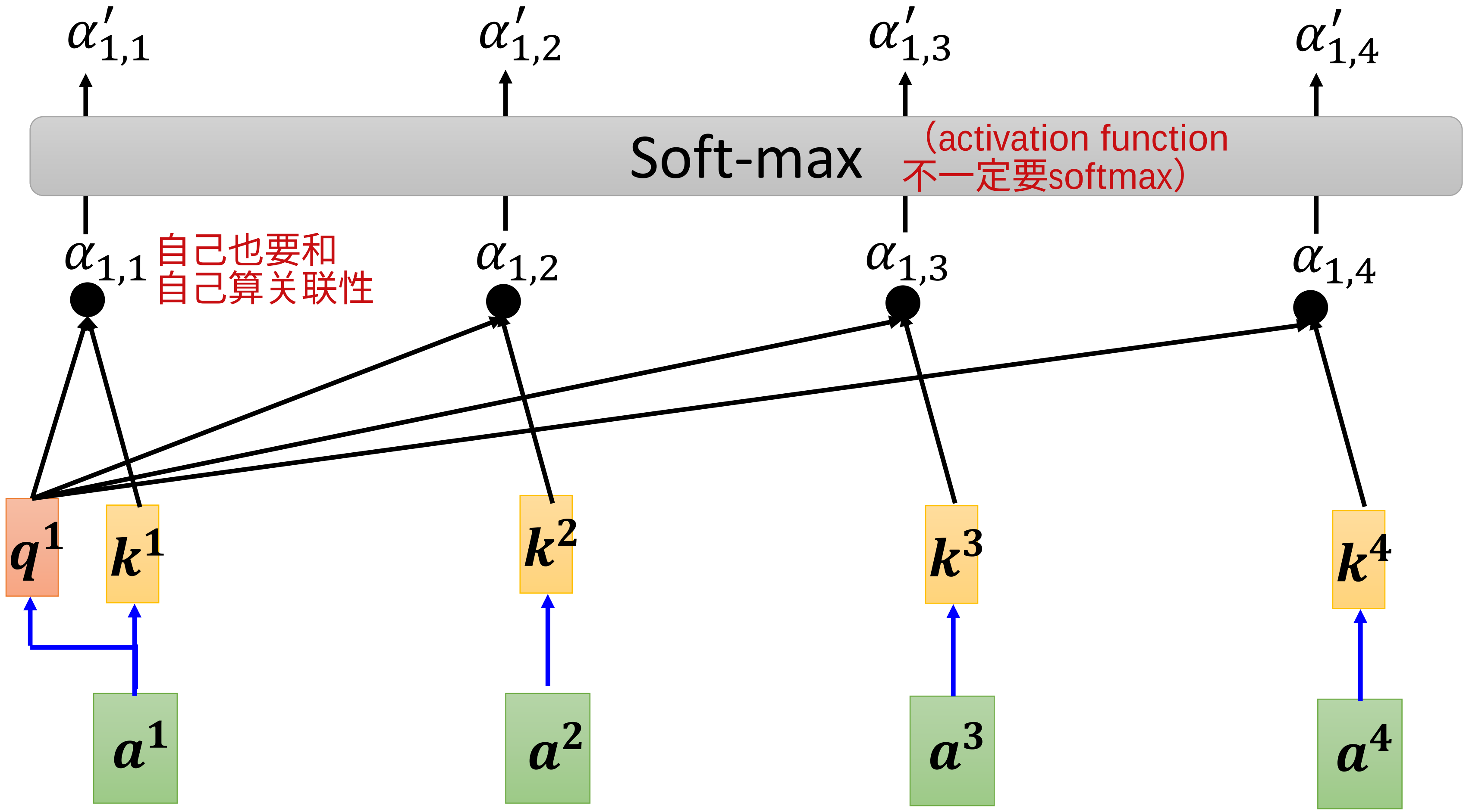
$$\alpha_{1,2} = \boldsymbol{q}^{1} \cdot \boldsymbol{k}^{2} \qquad \alpha_{1,3} = \boldsymbol{q}^{1} \cdot \boldsymbol{k}^{3} \qquad \alpha_{1,4} = \boldsymbol{q}^{1} \cdot \boldsymbol{k}^{4}$$



$$\boldsymbol{q}^{1} = W^{q}\boldsymbol{a}^{1} \qquad \boldsymbol{k}^{2} = W^{k}\boldsymbol{a}^{2} \qquad \boldsymbol{k}^{3} = W^{k}\boldsymbol{a}^{3} \qquad \boldsymbol{k}^{4} = W^{k}\boldsymbol{a}^{4}$$

Self-attention

$$\alpha'_{1,i} = exp(\alpha_{1,i})/\sum_j exp(\alpha_{1,j})$$



$\boldsymbol{q}^1 = W^q \boldsymbol{a}^1$ $\quad$ $\boldsymbol{k}^2 = W^k \boldsymbol{a}^2$ $\quad$ $\boldsymbol{k}^3 = W^k \boldsymbol{a}^3$ $\quad$ $\boldsymbol{k}^4 = W^k \boldsymbol{a}^4$

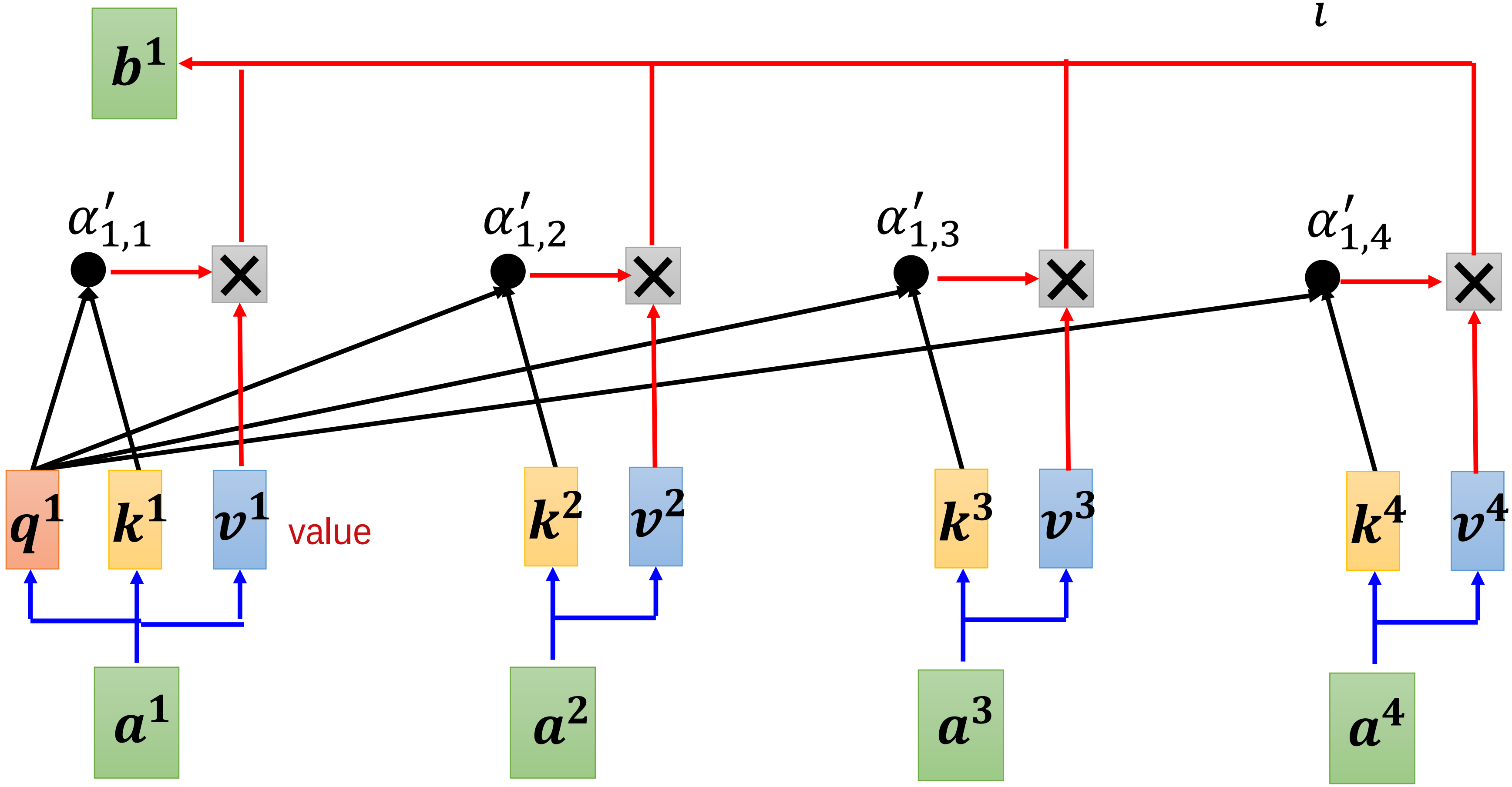
$\boldsymbol{k}^1 = W^k \boldsymbol{a}^1$

Self-attention

value

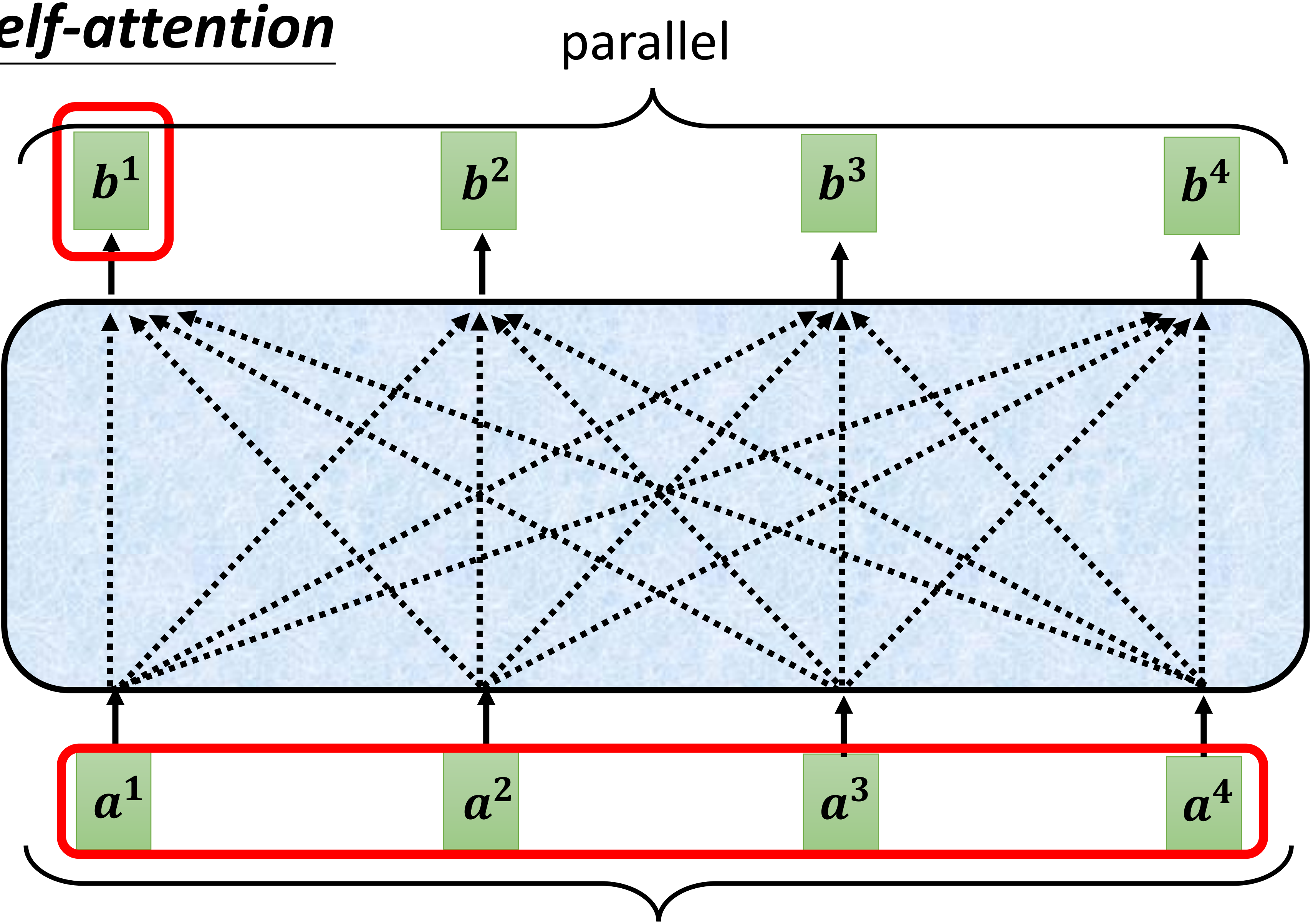
Extract information based on attention scores

$$\boldsymbol{b}^1 = \sum_i \alpha'_{1,i} \boldsymbol{v}^i$$



$\boldsymbol{v}^1 = \boxed{W^v} \boldsymbol{a}^1$ $\boldsymbol{v}^2 = W^v \boldsymbol{a}^2$ $\boldsymbol{v}^3 = W^v \boldsymbol{a}^3$ $\boldsymbol{v}^4 = W^v \boldsymbol{a}^4$

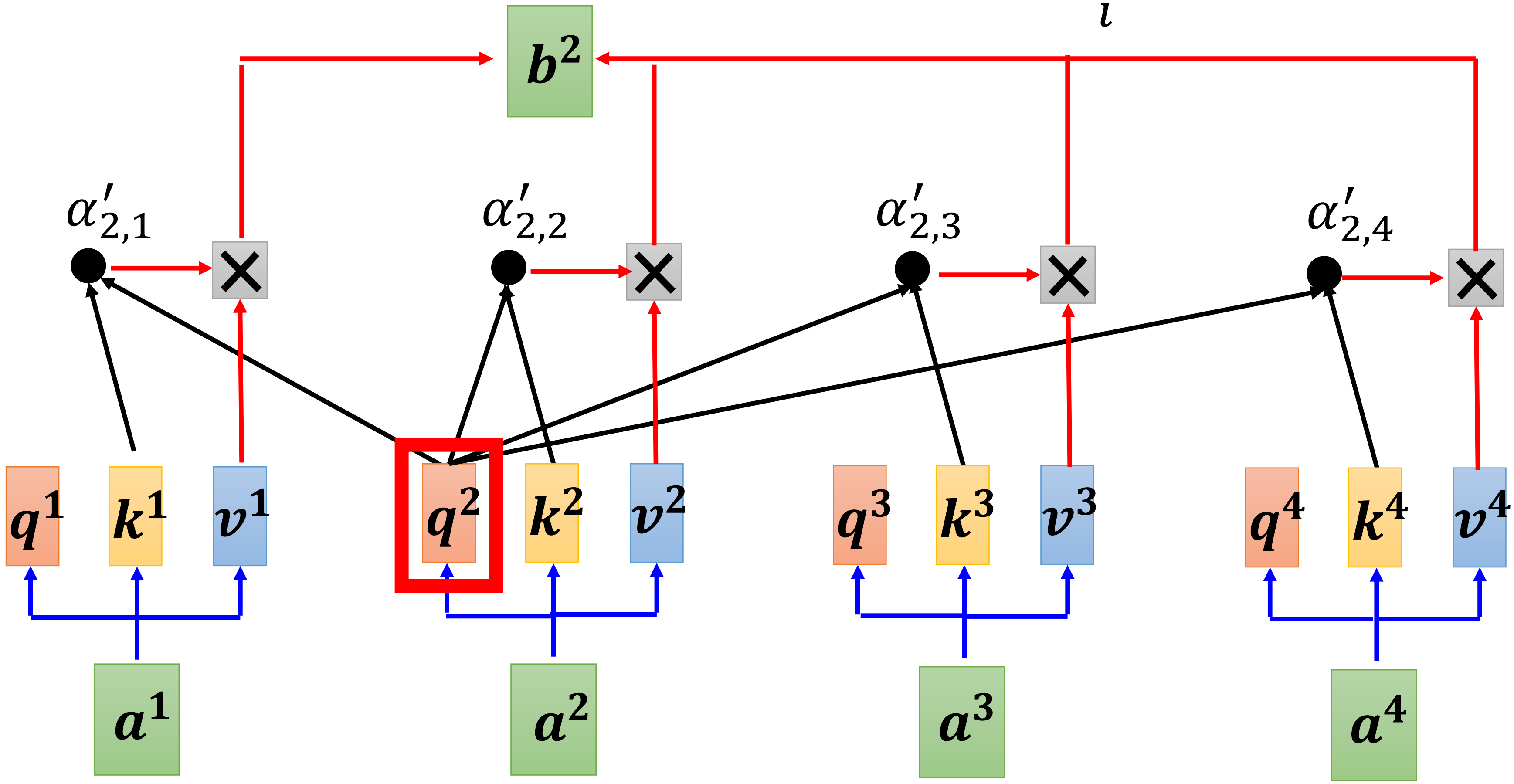
Self-attention



Can be either **input** or **a hidden layer**

Self-attention

$$b^2 = \sum_i \alpha'_{2,i} v^i$$



Self-attention

矩阵运算的角度：

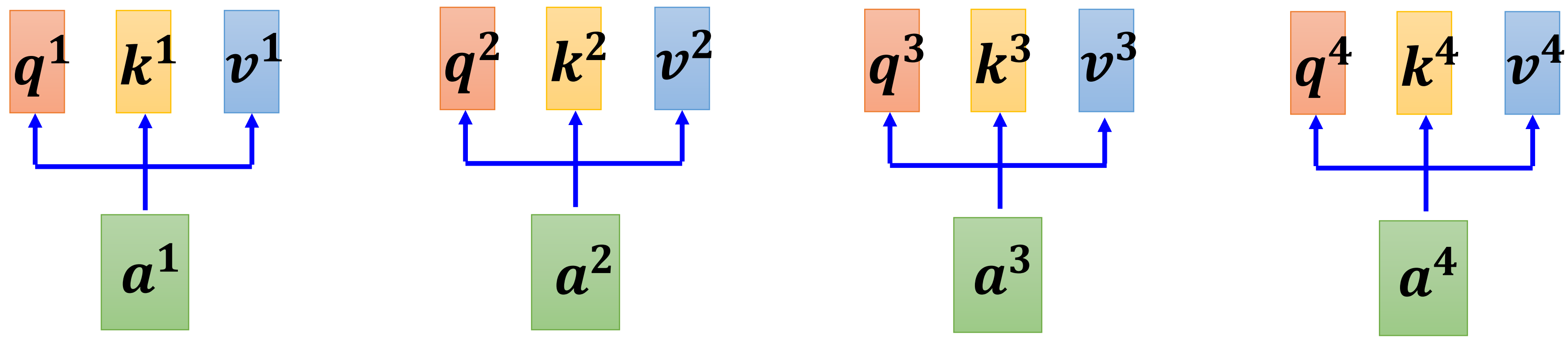
a → Q/K/V

$$q^i = W^q a^i \qquad \underbrace{\begin{bmatrix} q^1 & q^2 & q^3 & q^4 \end{bmatrix}}_{Q} = W^q \underbrace{\begin{bmatrix} a^1 & a^2 & a^3 & a^4 \end{bmatrix}}_{I}$$

network的参数，
会被learn出来

$$k^i = W^k a^i \qquad \underbrace{\begin{bmatrix} k^1 & k^2 & k^3 & k^4 \end{bmatrix}}_{K} = W^k \underbrace{\begin{bmatrix} a^1 & a^2 & a^3 & a^4 \end{bmatrix}}_{I}$$

$$v^i = W^v a^i \qquad \underbrace{\begin{bmatrix} v^1 & v^2 & v^3 & v^4 \end{bmatrix}}_{V} = W^v \underbrace{\begin{bmatrix} a^1 & a^2 & a^3 & a^4 \end{bmatrix}}_{I}$$



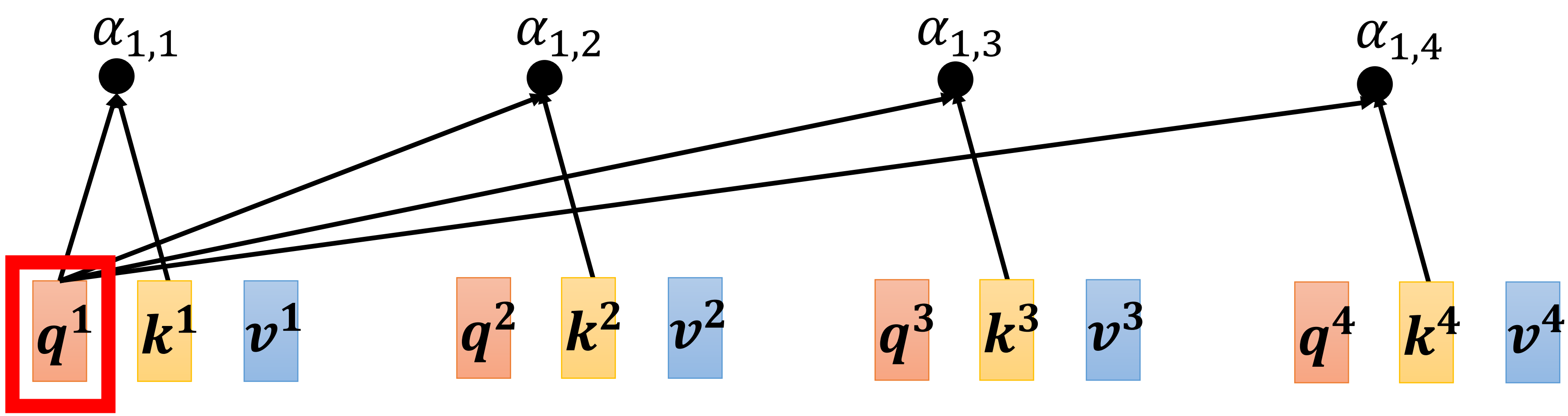
Self-attention

计算：Q/K → α

$$\alpha_{1,1} = k^1 q^1 \quad \alpha_{1,2} = k^2 q^1$$

$$\alpha_{1,3} = k^3 q^1 \quad \alpha_{1,4} = k^4 q^1$$

$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix} q^1$$



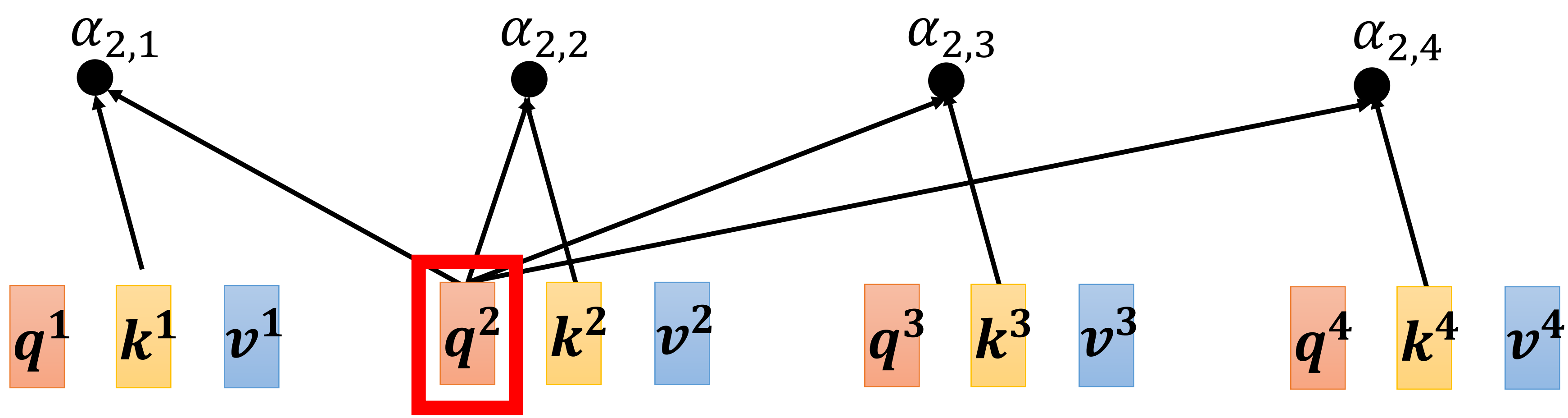
Self-attention

Q/K → α → α'

$$\alpha_{1,1} = \boldsymbol{k}^1 \boldsymbol{q}^1 \qquad \alpha_{1,2} = \boldsymbol{k}^2 \boldsymbol{q}^1$$

$$\alpha_{1,3} = \boldsymbol{k}^3 \boldsymbol{q}^1 \qquad \alpha_{1,4} = \boldsymbol{k}^4 \boldsymbol{q}^1$$

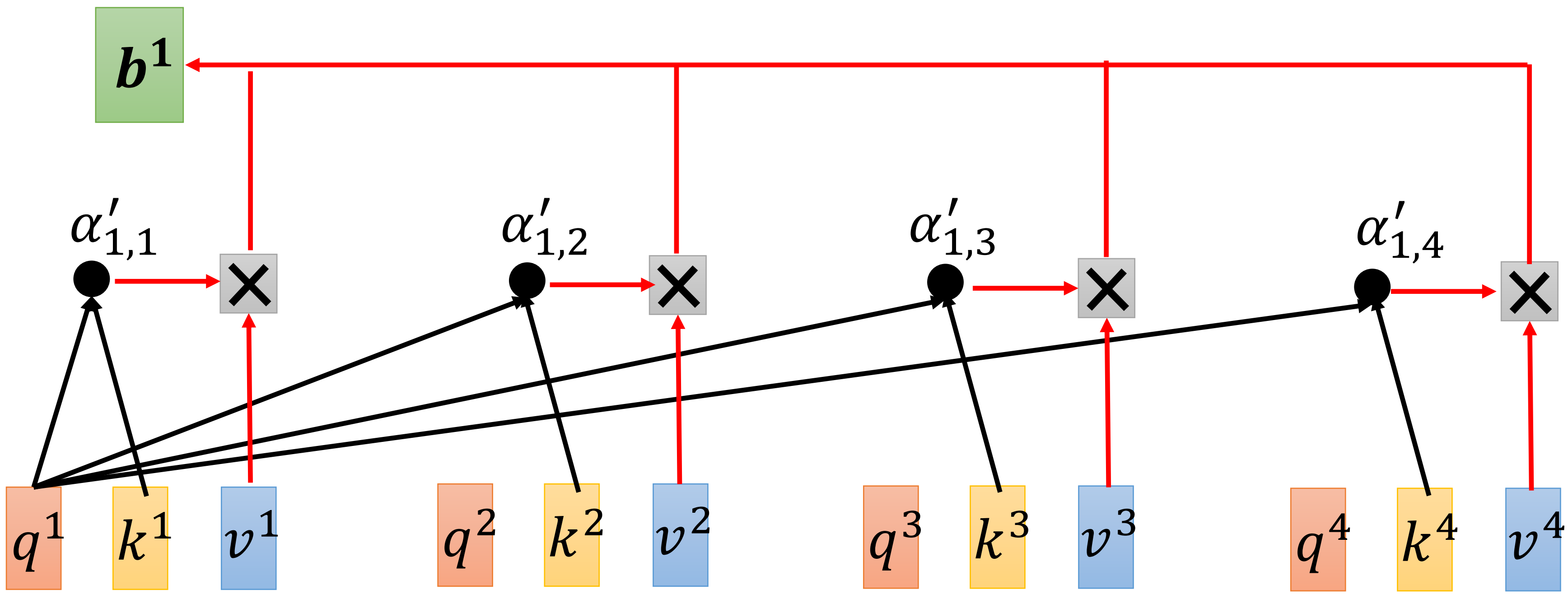
$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}^1 \\ \boldsymbol{k}^2 \\ \boldsymbol{k}^3 \\ \boldsymbol{k}^4 \end{bmatrix} \boldsymbol{q}^1$$



$$\underbrace{\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}}_{A'} \xleftarrow{\text{softmax}} \underbrace{\begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1} \\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2} \\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3} \\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \boldsymbol{k}^1 \\ \boldsymbol{k}^2 \\ \boldsymbol{k}^3 \\ \boldsymbol{k}^4 \end{bmatrix}}_{K^T} \underbrace{\begin{bmatrix} \boldsymbol{q}^1 & \boldsymbol{q}^2 & \boldsymbol{q}^3 & \boldsymbol{q}^4 \end{bmatrix}}_{Q}$$

Self-attention

V/α' → b



$$\underbrace{b^1\ b^2\ b^3\ b^4}_{O} = \underbrace{v^1\ v^2\ v^3\ v^4}_{V} \underbrace{\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}}_{A'}$$

Self-attention

$$Q = W^q \, I$$

$$K = W^k \, I$$

$$V = W^v \, I$$

Parameters to be learned

只有这三个参数需要学习获得

$$A' \leftarrow A = K^T Q$$

Attention Matrix

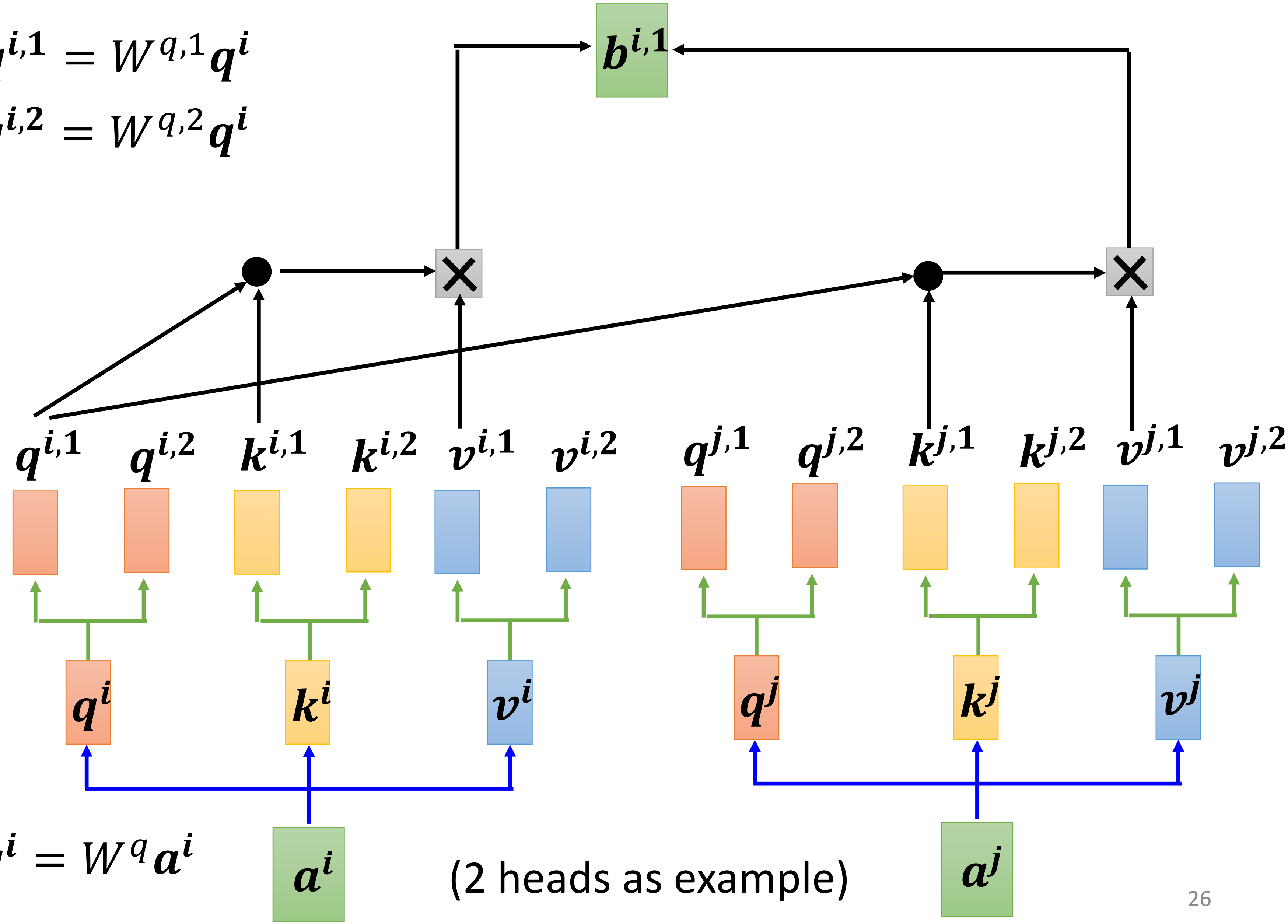
$$O = V A'$$

Multi-head Self-attention

Different types of relevance

$q^{i,1} = W^{q,1} q^i$

$q^{i,2} = W^{q,2} q^i$



$q^i = W^q a^i$

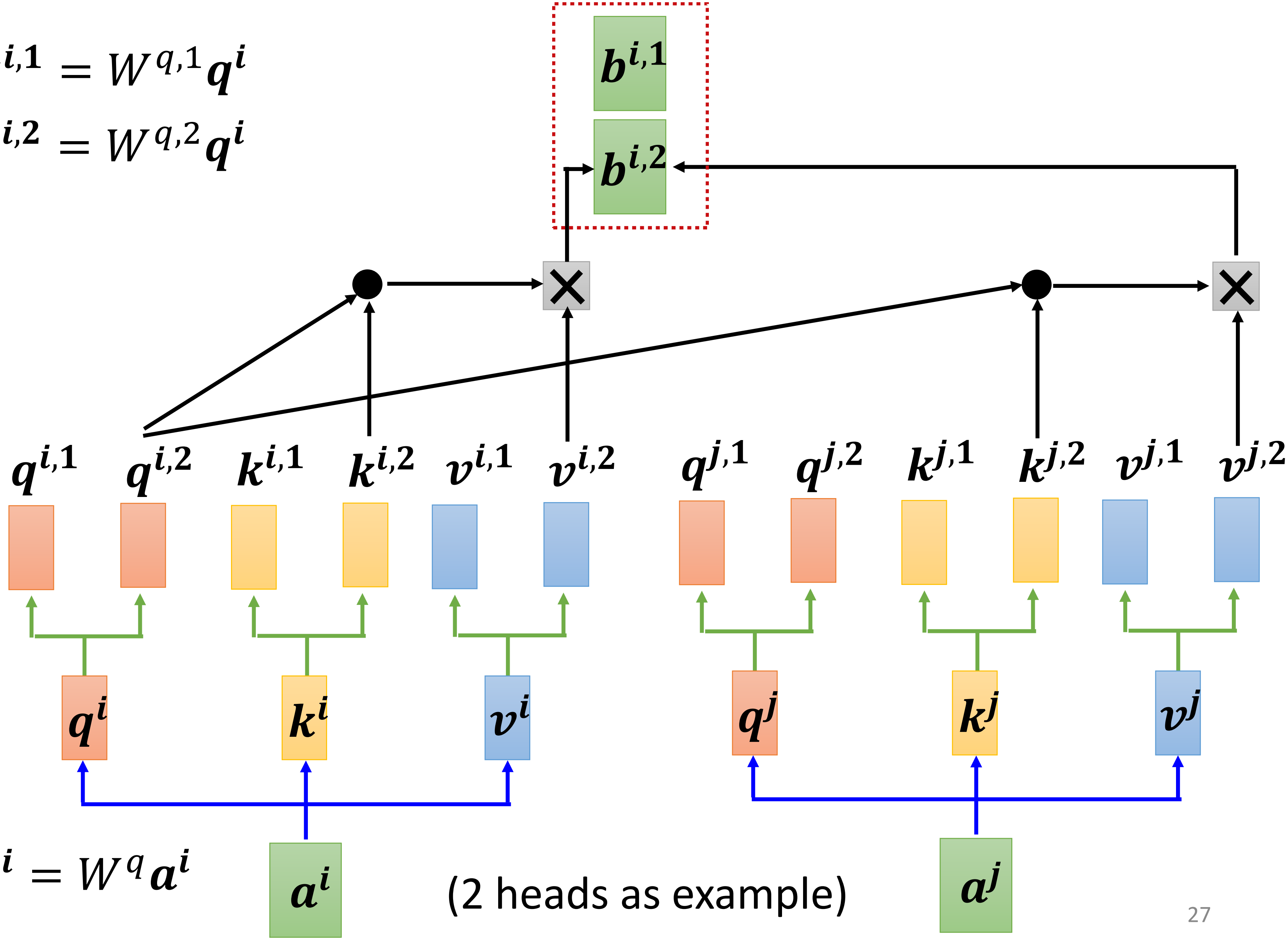
(2 heads as example)

Multi-head Self-attention

Different types of relevance

$$q^{i,1} = W^{q,1} q^i$$

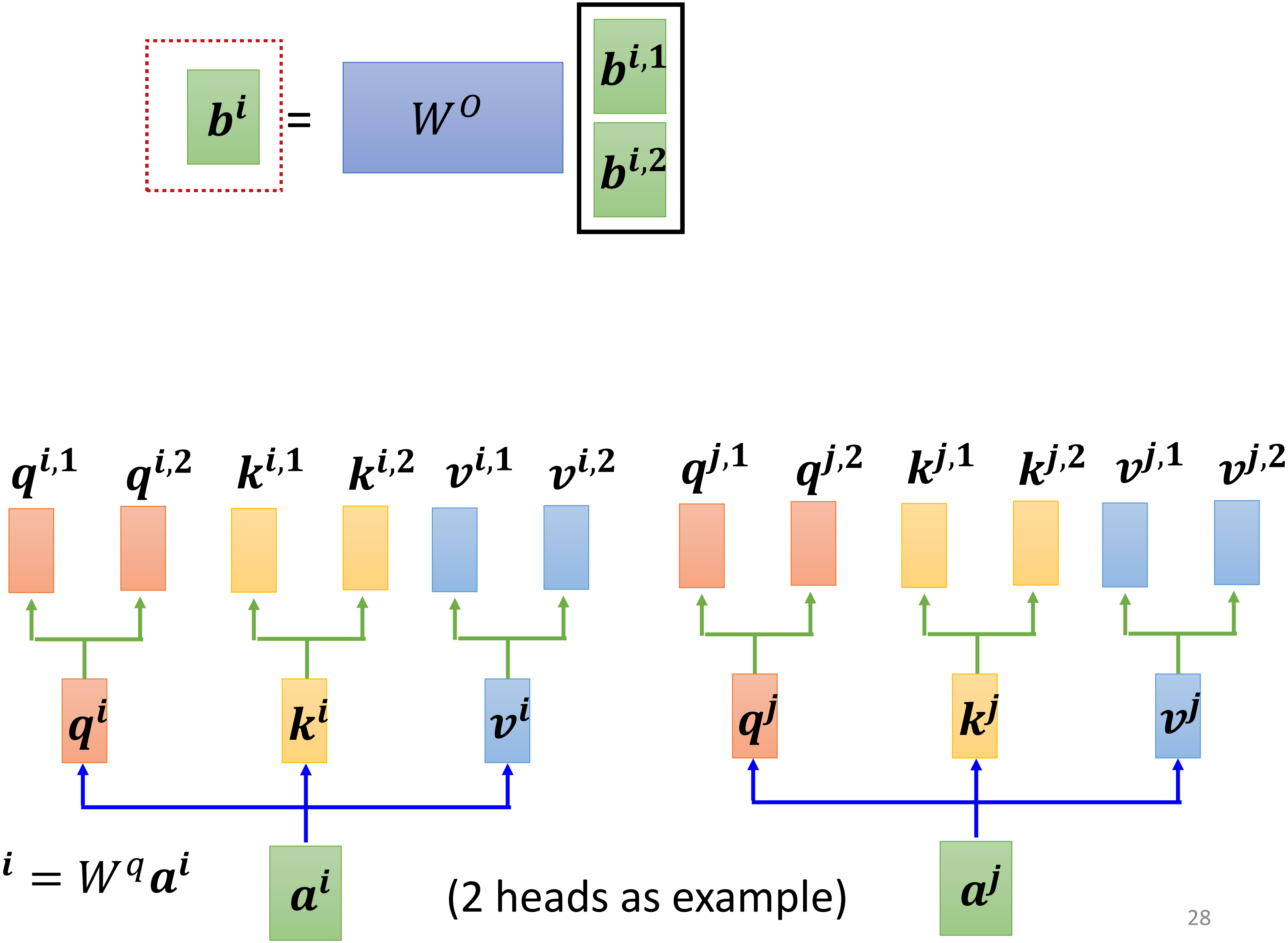
$$q^{i,2} = W^{q,2} q^i$$



$$q^i = W^q a^i$$

(2 heads as example)

Multi-head Self-attention Different types of relevance



$$q^i = W^q a^i$$

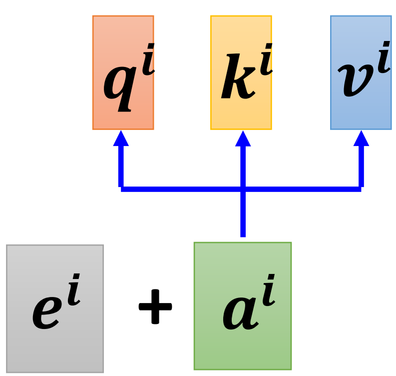
(2 heads as example)

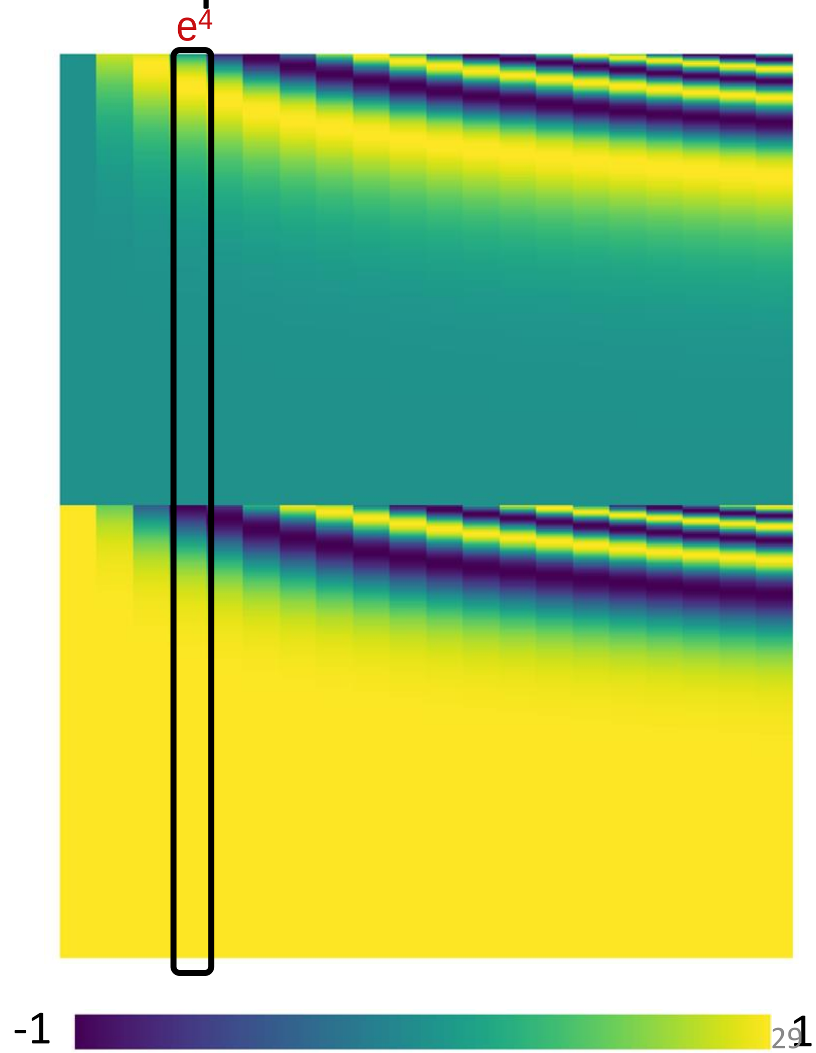
Positional Encoding

Attention is all you need里
positional encoding的设定

Each column represents a positional vector e^i

- No position information in self-attention.
- Each position has a unique positional vector e^i
- **hand-crafted**
- **learned from data**



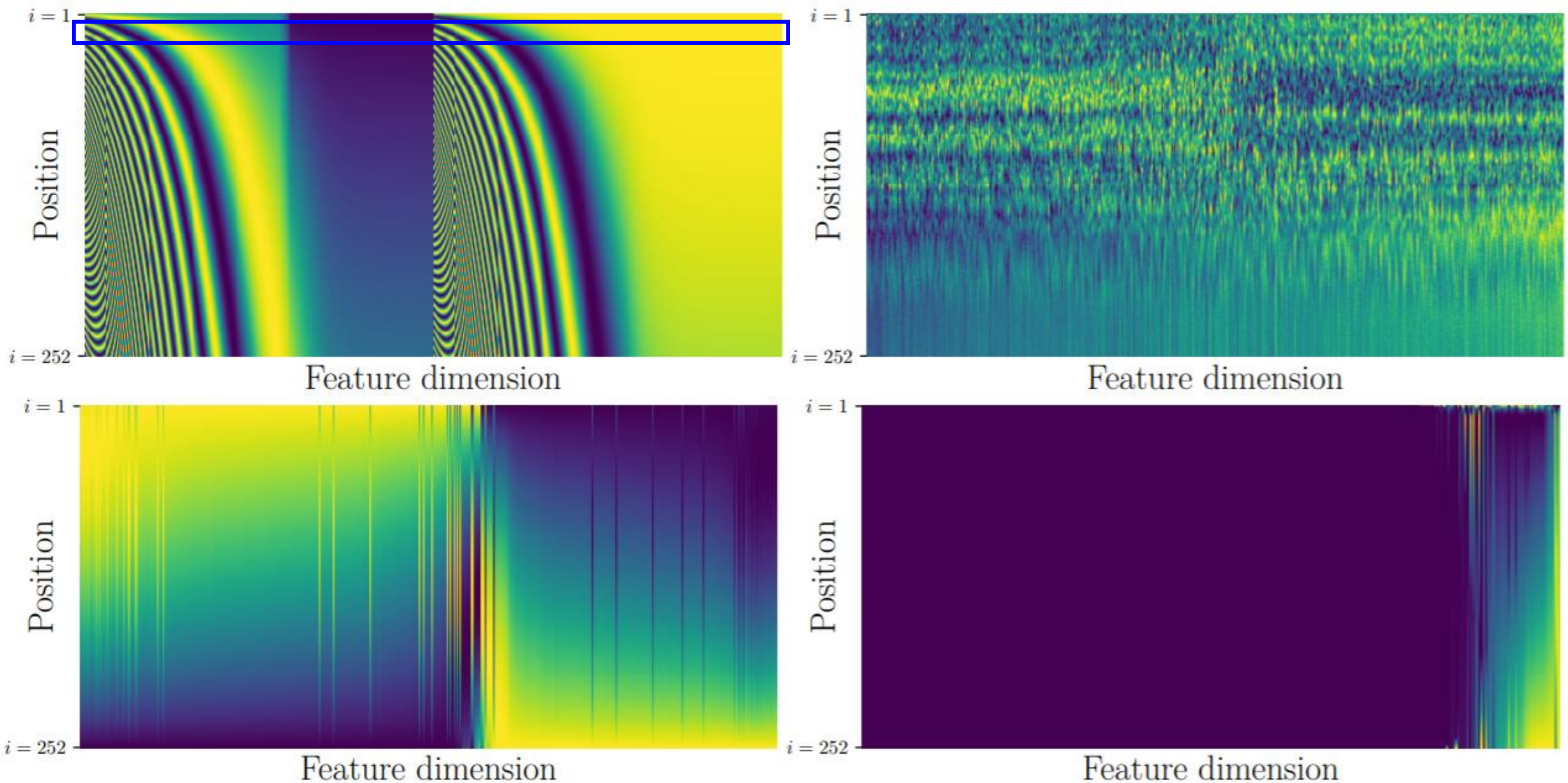


https://arxiv.org/abs/2003.09229

Table 1. Comparing position representation methods

Methods	Inductive	Data-Driven	Parameter Efficient
Sinusoidal (Vaswani et al., 2017)	✓	✗	✓
Embedding (Devlin et al., 2018)	✗	✓	✗
Relative (Shaw et al., 2018)	✗	✓	✓
This paper	✓	✓	✓

(a) Sinusoidal (b) Position embedding



(c) FLOATER (d) RNN

Many applications …

Transformer

https://arxiv.org/abs/1706.03762

BERT

https://arxiv.org/abs/1810.04805

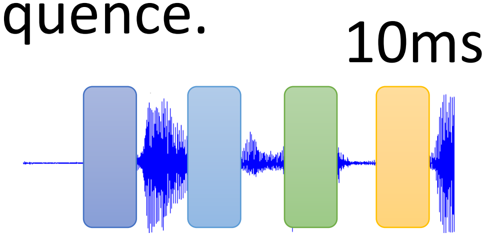
Widely used in Natural Langue Processing (NLP)!

https://arxiv.org/abs/1910.12977

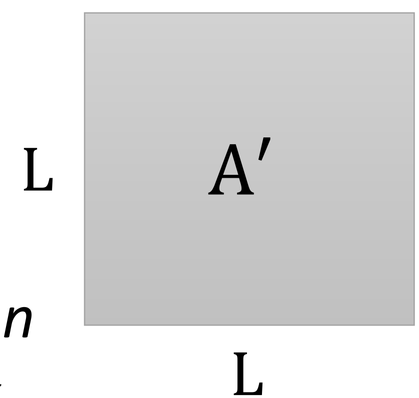
Self-attention for Speech 语音上的应用

Speech is a very long vector sequence.

sequence很长

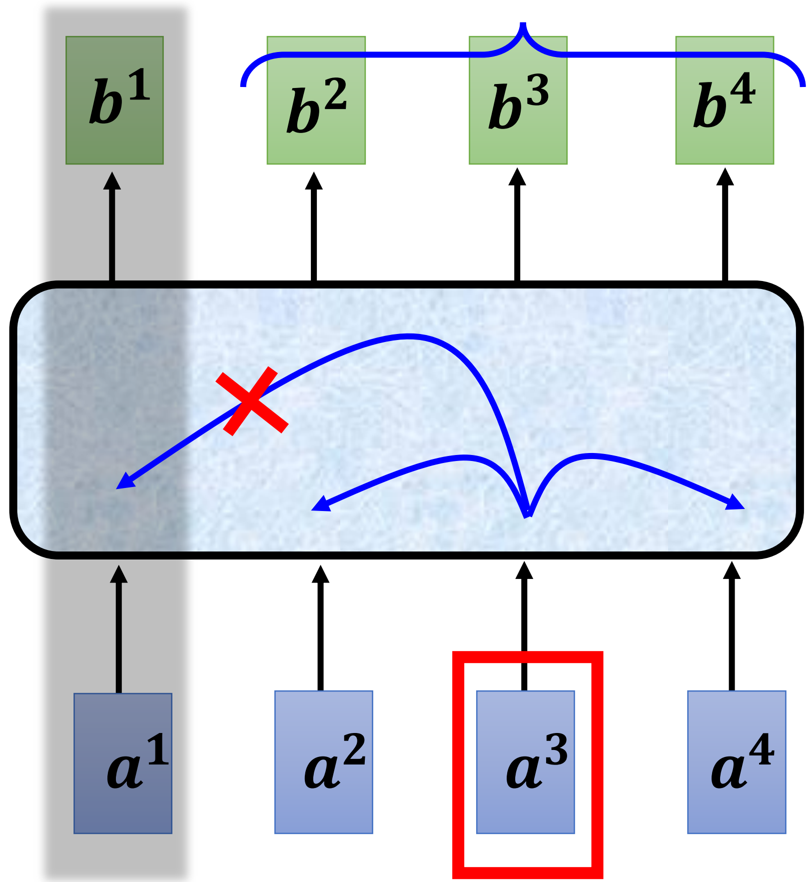


If input sequence is length L



Attention Matrix

Attention in a range

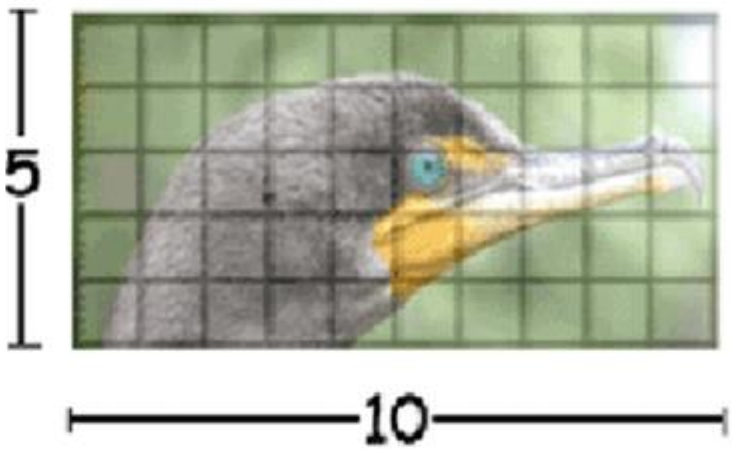


Truncated Self-attention

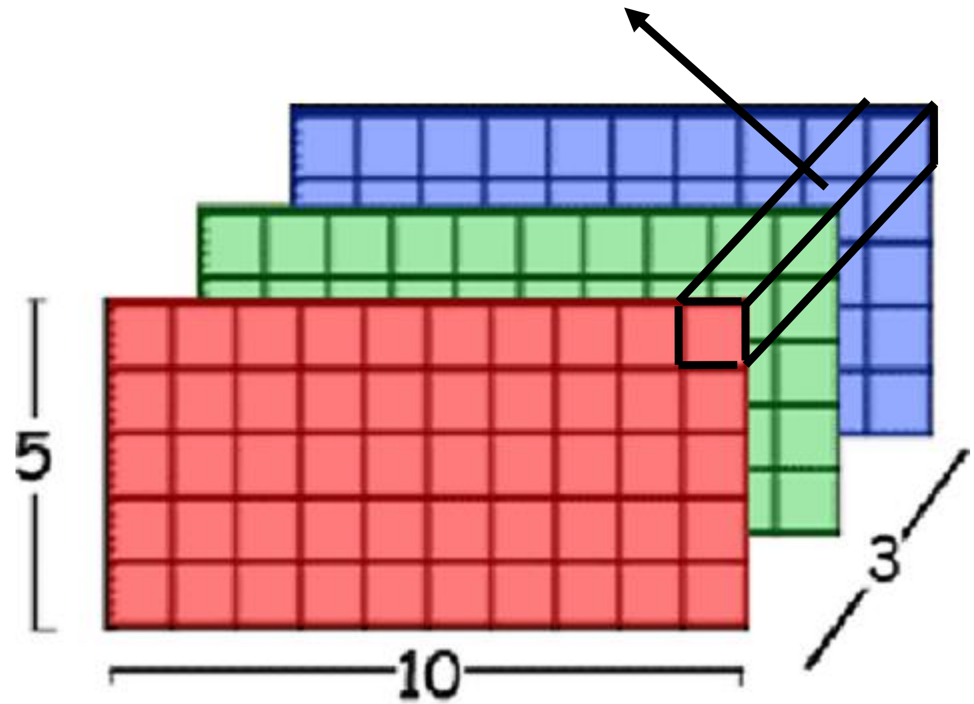
小范围内做self-attention

Self-attention for Image 影像处理上的应用

An **image** can also be considered as a **vector set**.

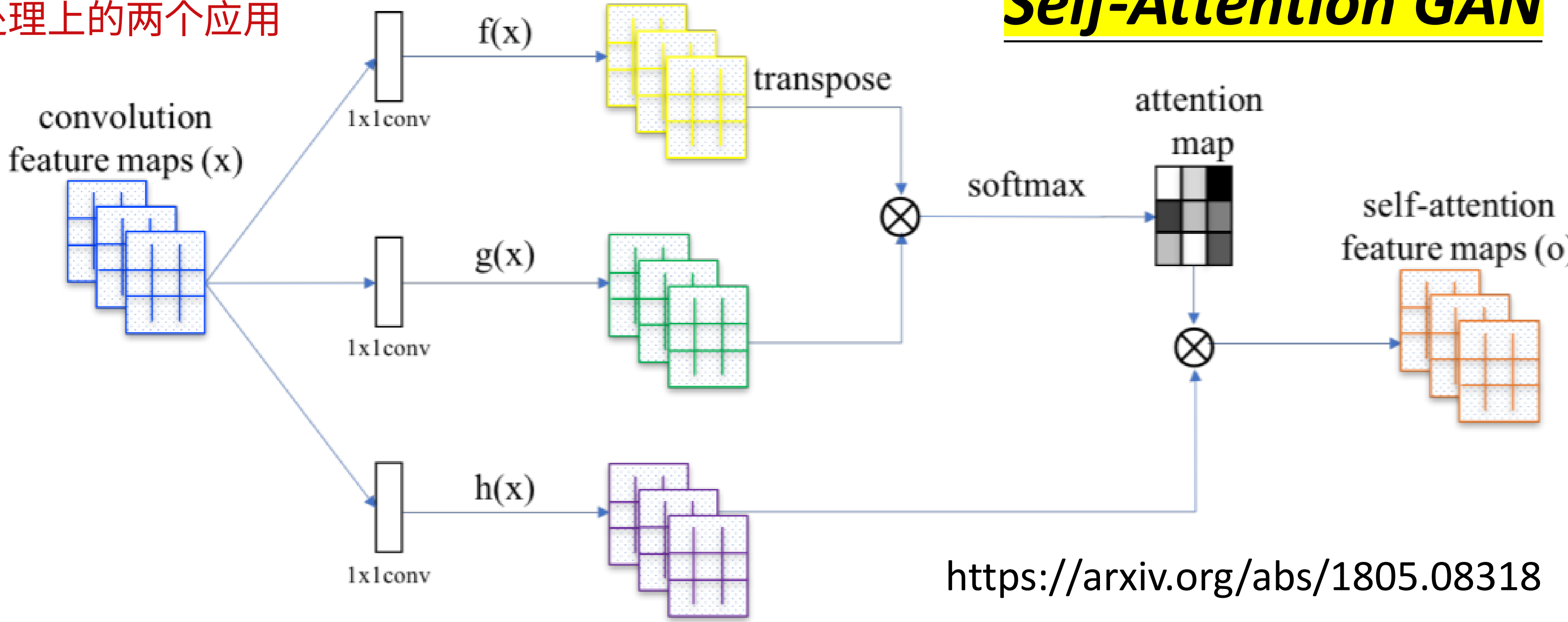


This is a vector.



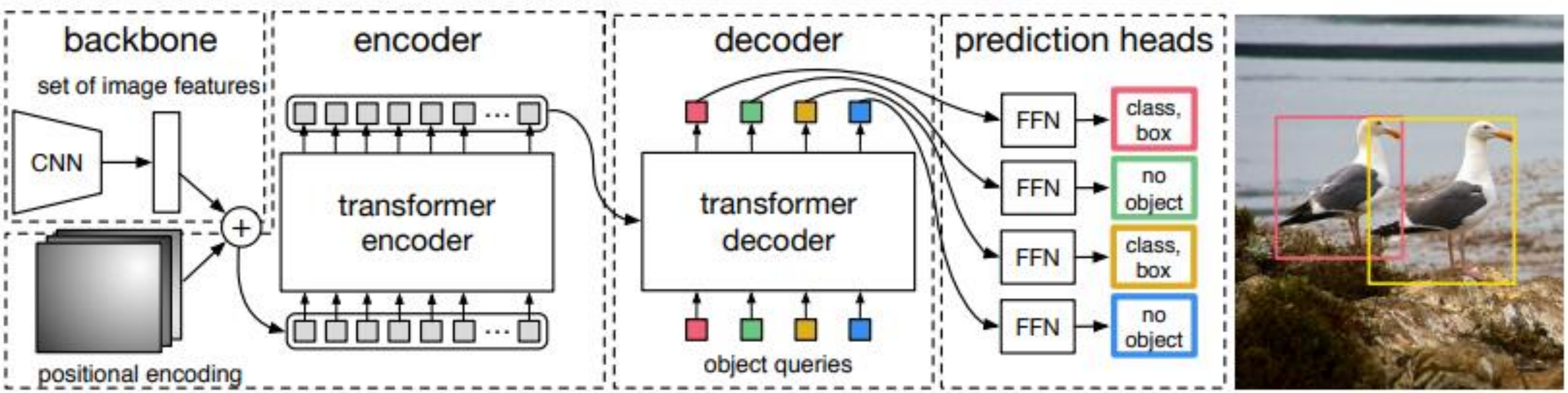
Source of image: https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184 self-attention在影像处理上的两个应用

Self-Attention GAN



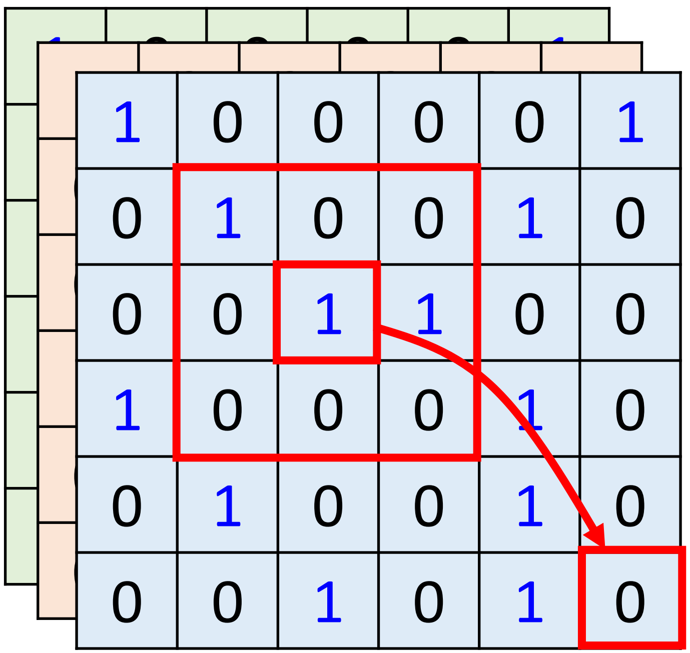
https://arxiv.org/abs/1805.08318

DEtection Transformer (DETR)



https://arxiv.org/abs/2005.12872

Self-attention v.s. CNN



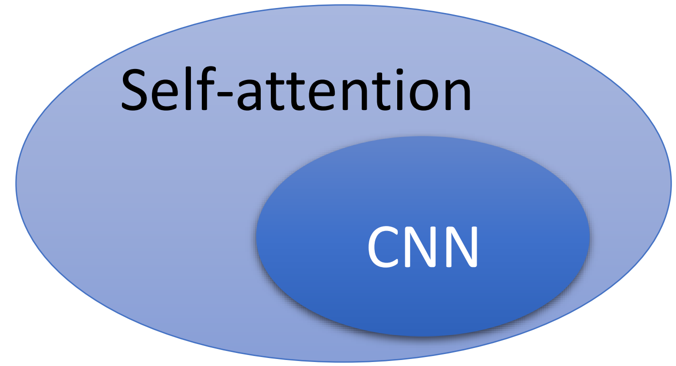
CNN: self-attention that can only attends in a receptive field

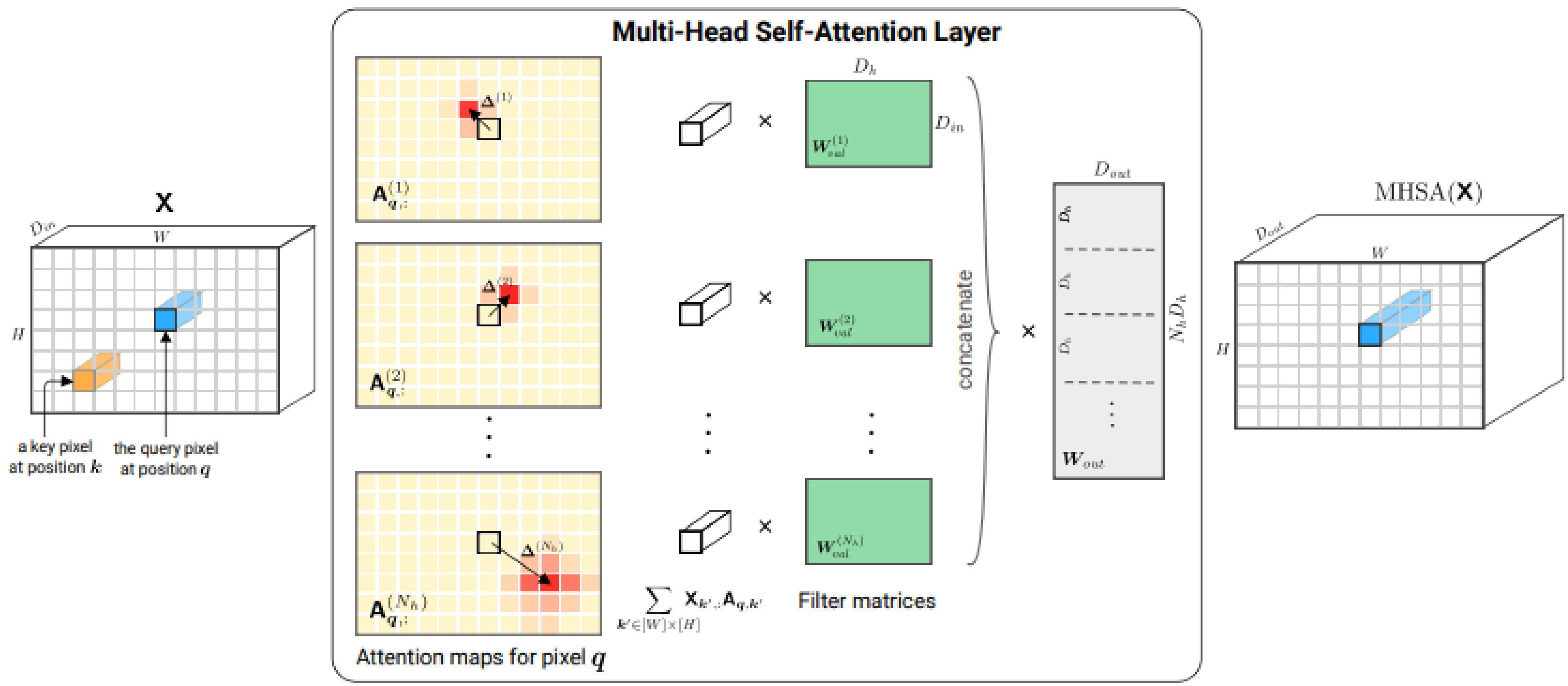
- CNN is simplified self-attention.

Self-attention: CNN with learnable receptive field

- Self-attention is the complex version of CNN.

Self-attention v.s. CNN

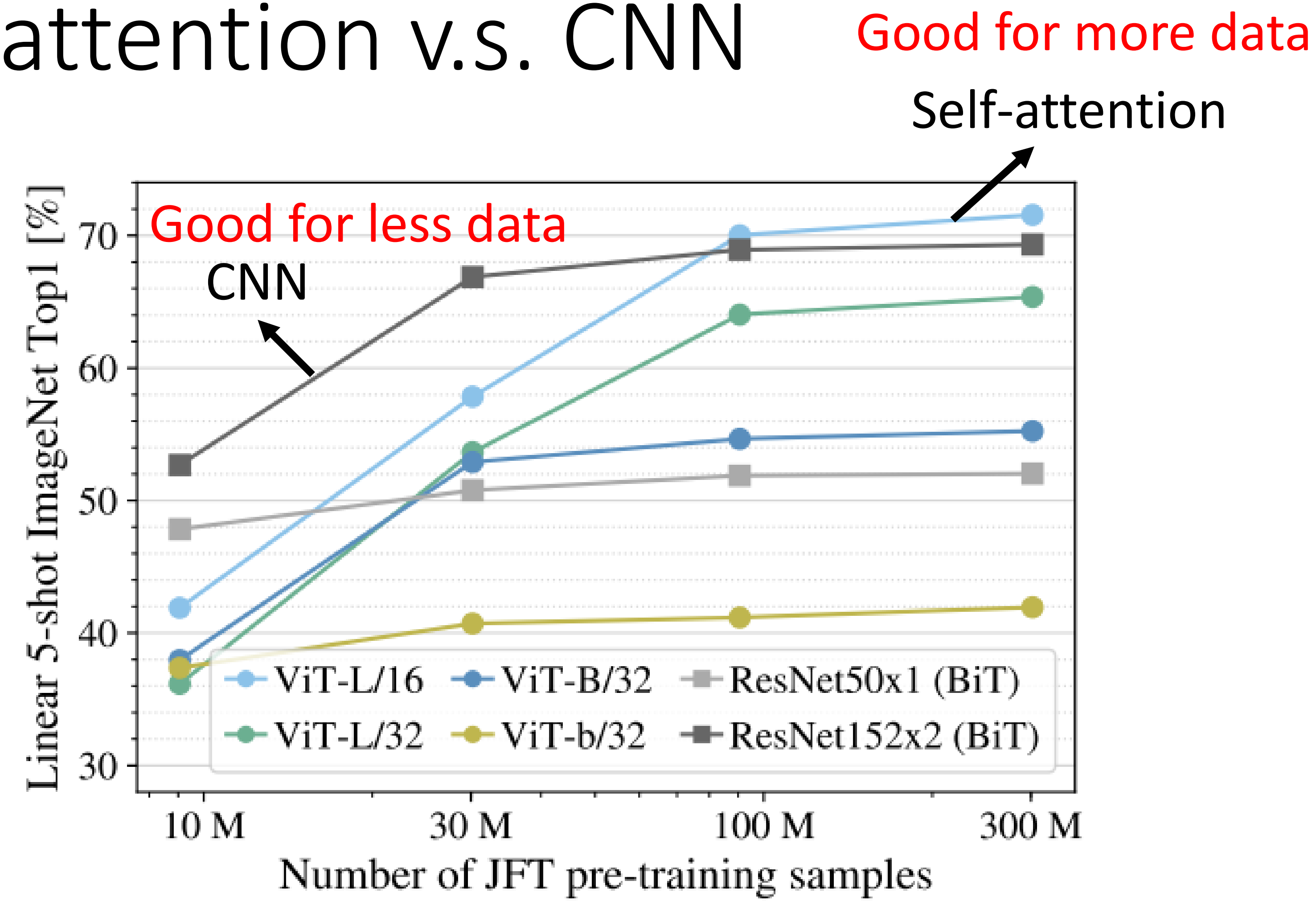




On the Relationship between Self-Attention and Convolutional Layers

https://arxiv.org/abs/1911.03584

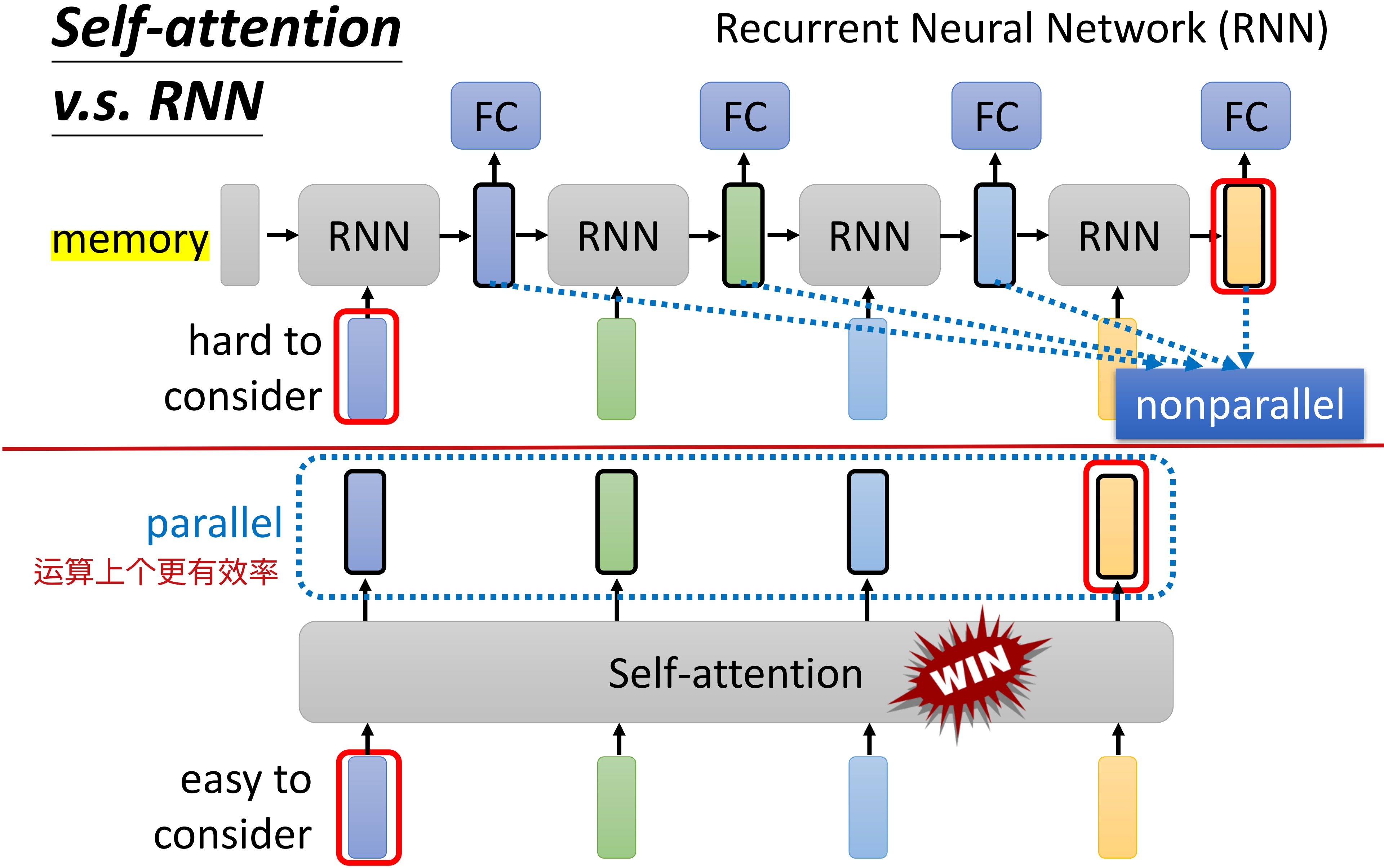
Self-attention v.s. CNN



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

https://arxiv.org/pdf/2010.11929.pdf

Self-attention v.s. RNN



Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention

https://arxiv.org/abs/2006.16236

To learn more about RNN ……

https://youtu.be/xCGidAeyS4M

(in Mandarin)

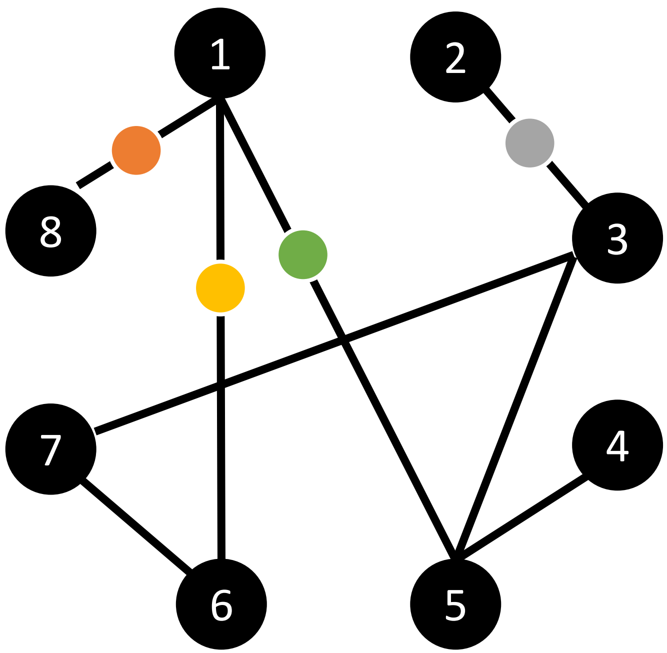
2017年ML课RNN

https://youtu.be/Jjy6ER0bHv8

(in English)

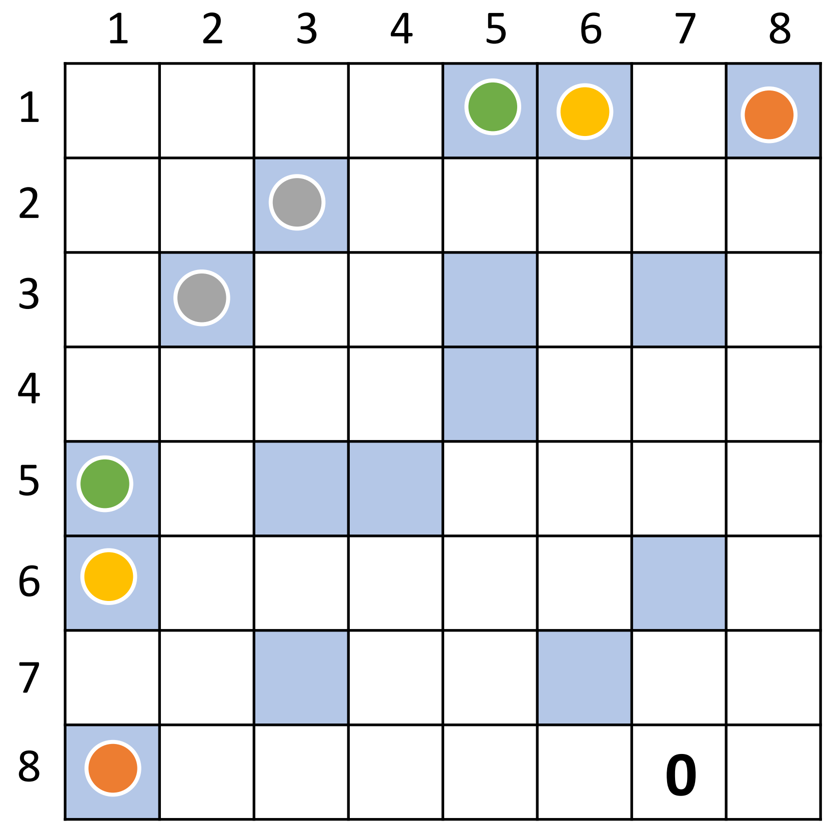
Graph可以被看作是set of vectors

Self-attention for Graph



Consider **edge**: only attention to connected nodes

Attention Matrix



This is one type of **Graph Neural Network (GNN)**.

Self-attention for Graph

- To learn more about GNN …

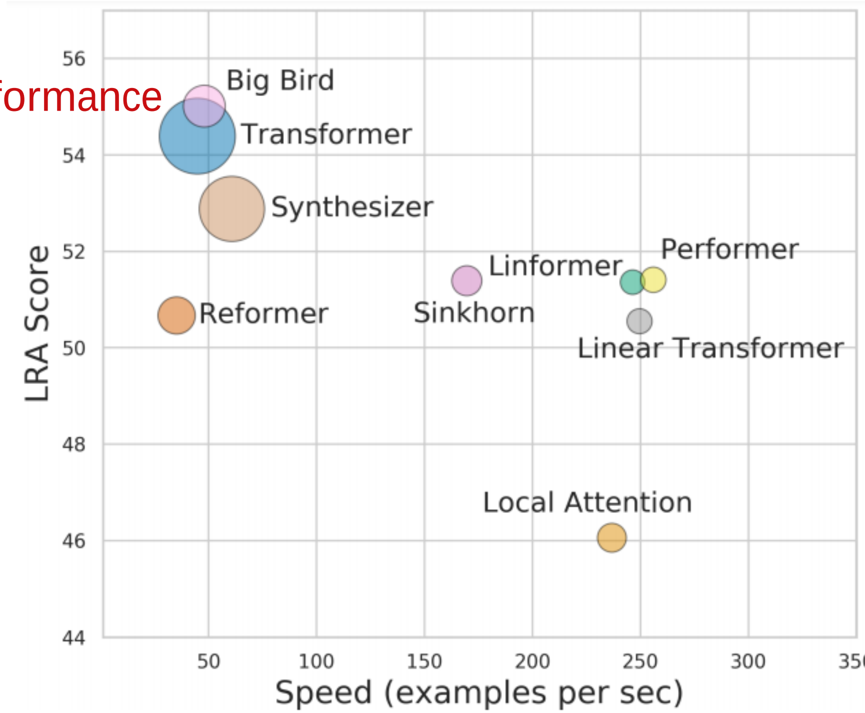
https://youtu.be/eybCCtNKwzA
(in Mandarin)

https://youtu.be/M9ht8vsVEw8
(in Mandarin)

To Learn More ...

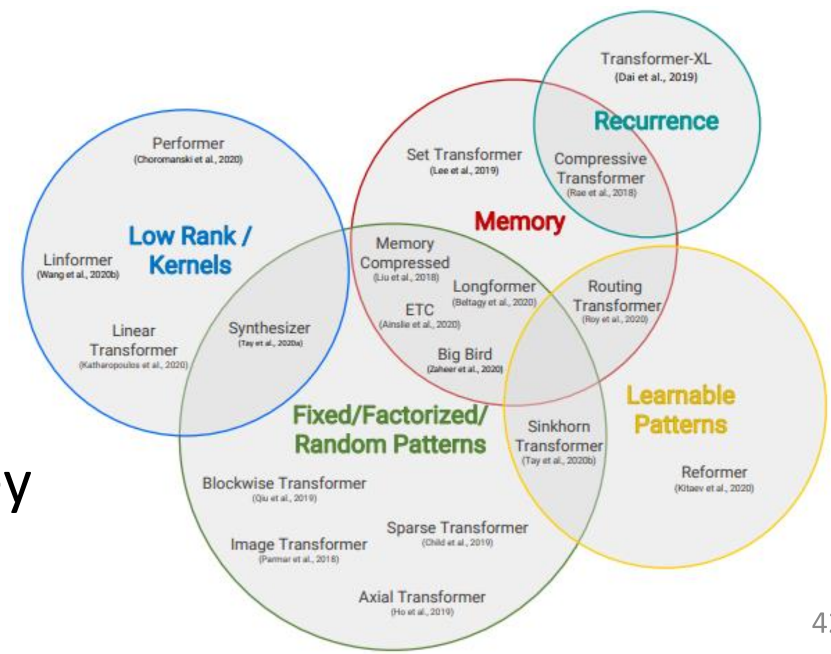
self-attention的变形
self-attention的运算量很大，
怎么较少运算量是未来的一个课题。

performance



Long Range Arena: A Benchmark for Efficient Transformers

https://arxiv.org/abs/2011.04006



Efficient Transformers: A Survey

https://arxiv.org/abs/2009.06732


Q&A