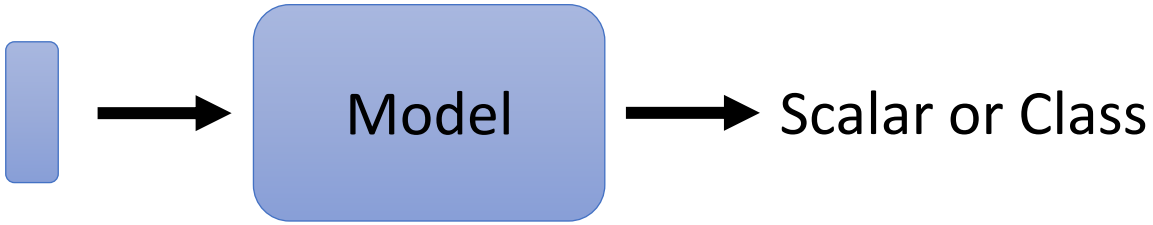
Self-attention

Hung-yi Lee
李宏毅

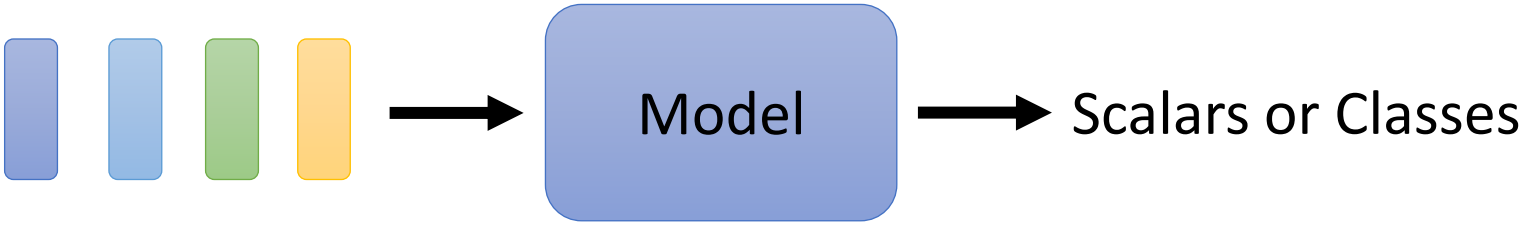
是一種nn架構
（cnn也是一種nn架構）

Sophisticated Input

- Input is **a vector**



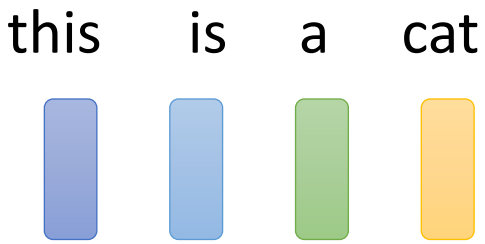
- Input is **a set of vectors**



(may change length)

若input是一連串的vector
且數量不固定

Vector Set as Input



假設詞彙倆倆之間沒有關係

One-hot Encoding

apple = [1 0 0 0 0 .....]

bag = [0 1 0 0 0 .....]

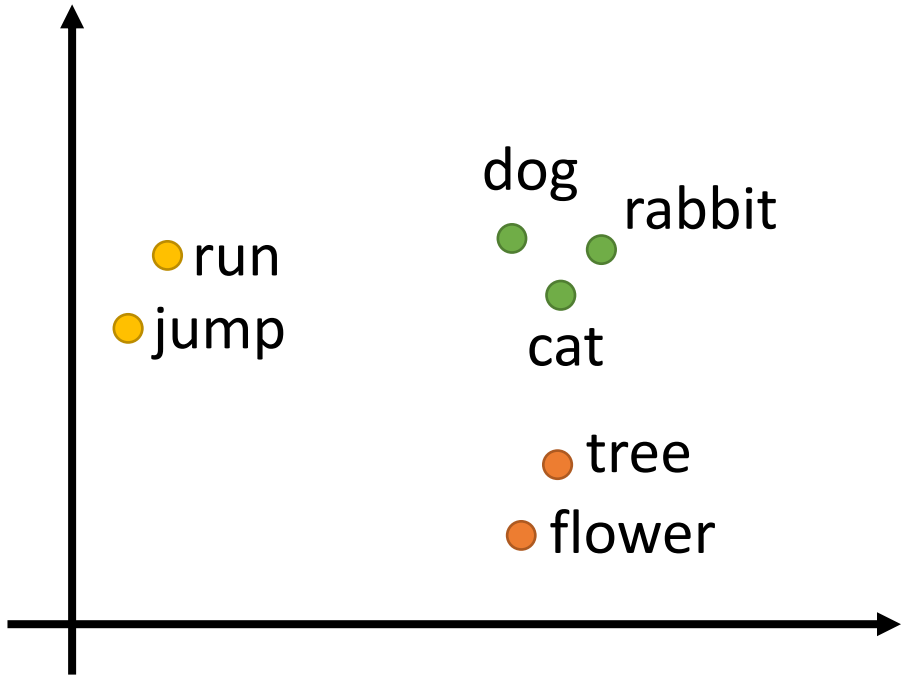
cat = [0 0 1 0 0 .....]

dog = [0 0 0 1 0 .....]

elephant = [0 0 0 0 1 .....]

可以呈現詞彙之間的關係

Word Embedding



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

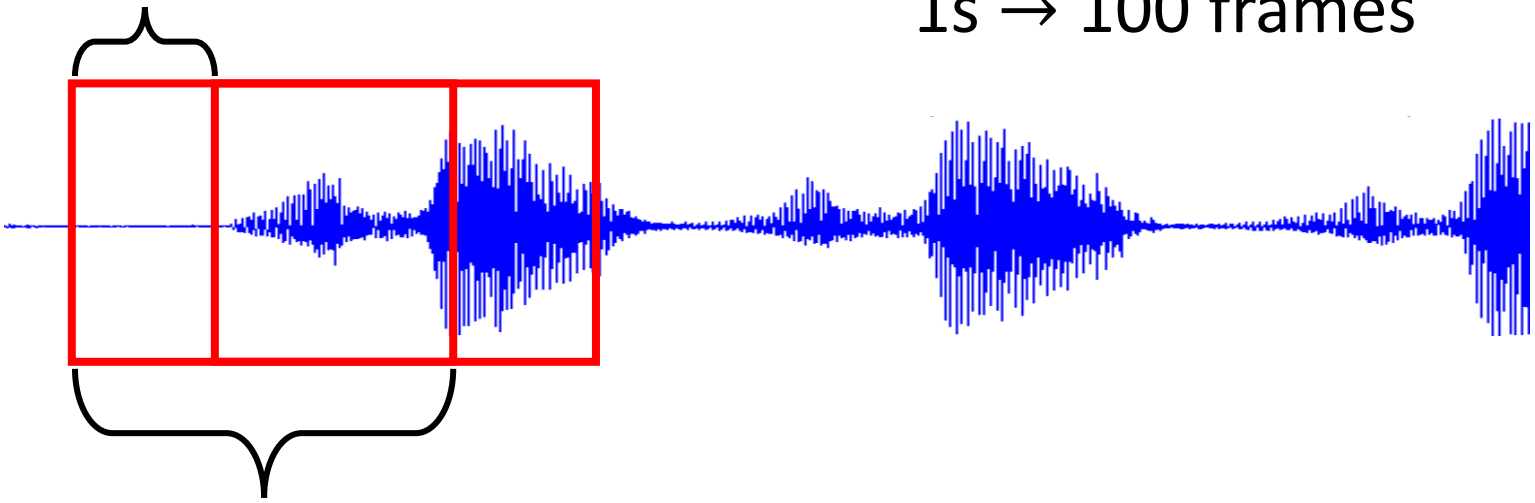
word embedding的課程

Vector Set as Input

為什麼是25ms/10ms？
古聖先賢的經驗累積

10ms 相當於cnn的stride

1s → 100 frames



25ms 每個window 25ms，再將他變成一個frame

frame

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**)

Each profile is a vector

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

Vector Set as Input

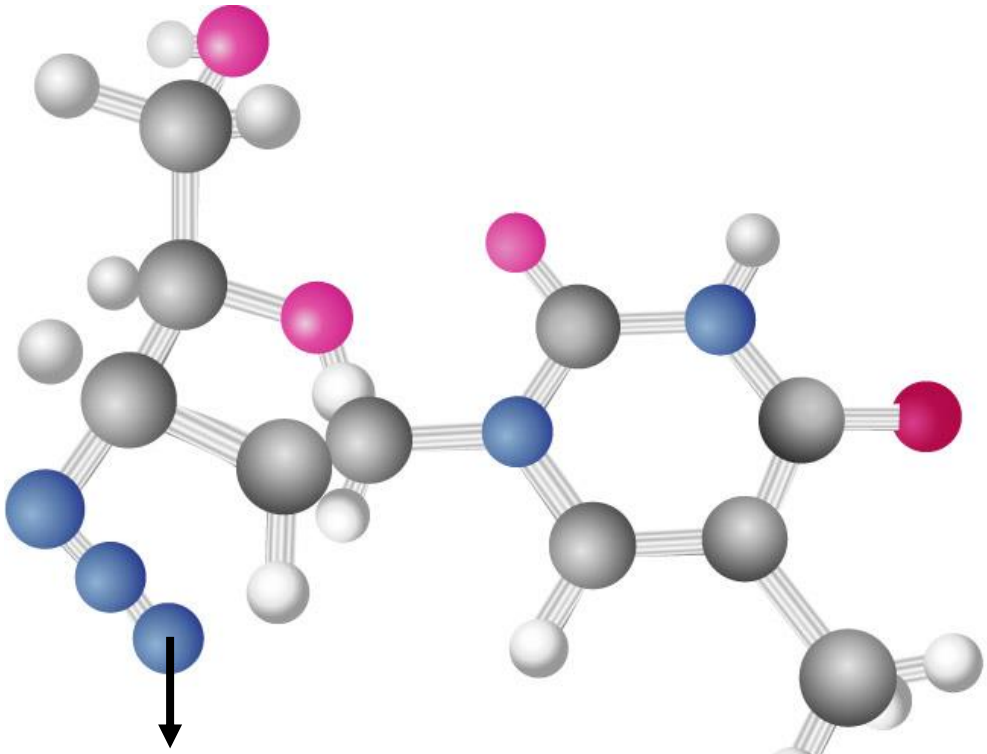
- Graph is also a set of vectors (consider each **node** as **a vector**)

H = [1 0 0 0 0 ……]

C = [0 1 0 0 0 ……]

O = [0 0 1 0 0 ……]

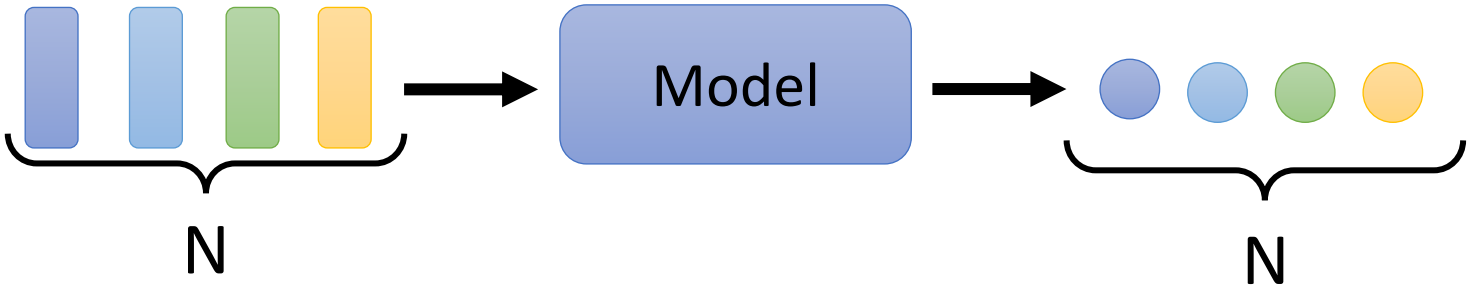
⋮



One-hot vector

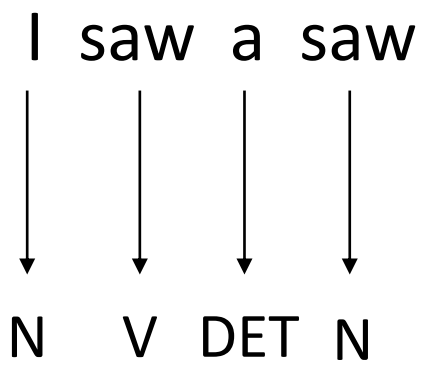
What is the output?

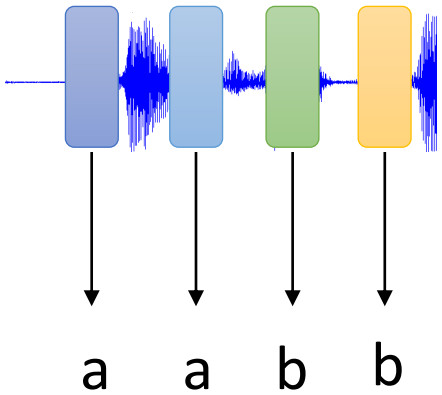
- Each vector has a label.

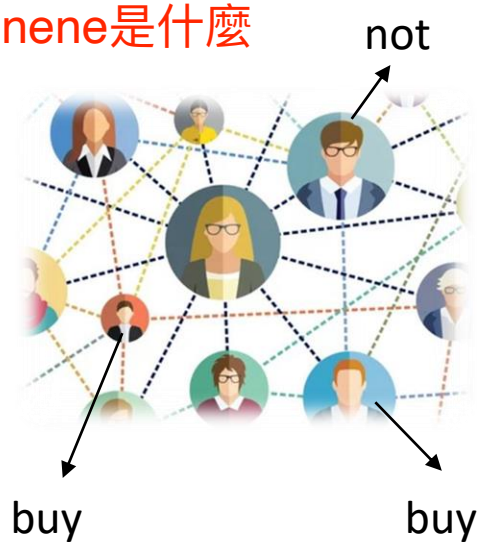


Example Applications

作業二某種程度上也算是
要判斷每個聲音訊號的phonene是什麼







POS tagging

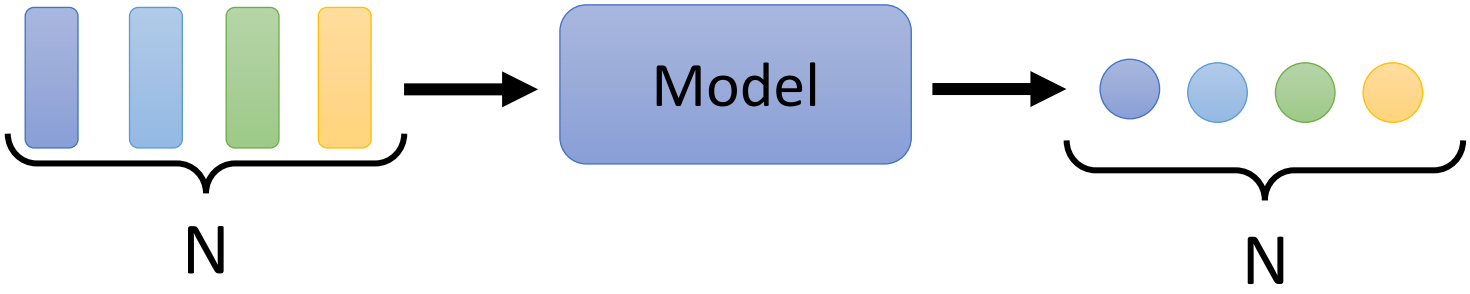
例如詞性標註，就是每個vector給一個答案

HW2

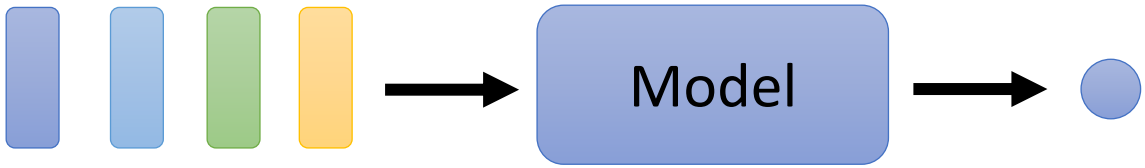
去預測每個node會不會去買某樣東西

What is the output?

- Each vector has a label.



- The whole sequence has a label.



給一個化學分子的graph，輸出他類別

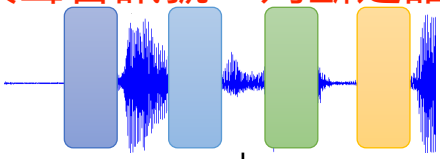
Example Applications

給一段聲音訊號，判斷是誰在講話

this is good

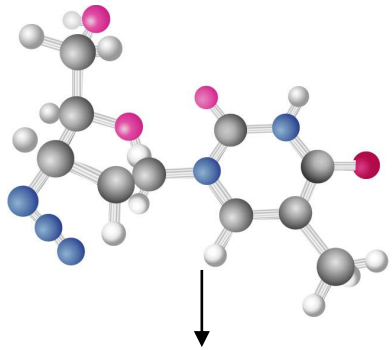
Sentiment analysis

positive



HW4

speaker



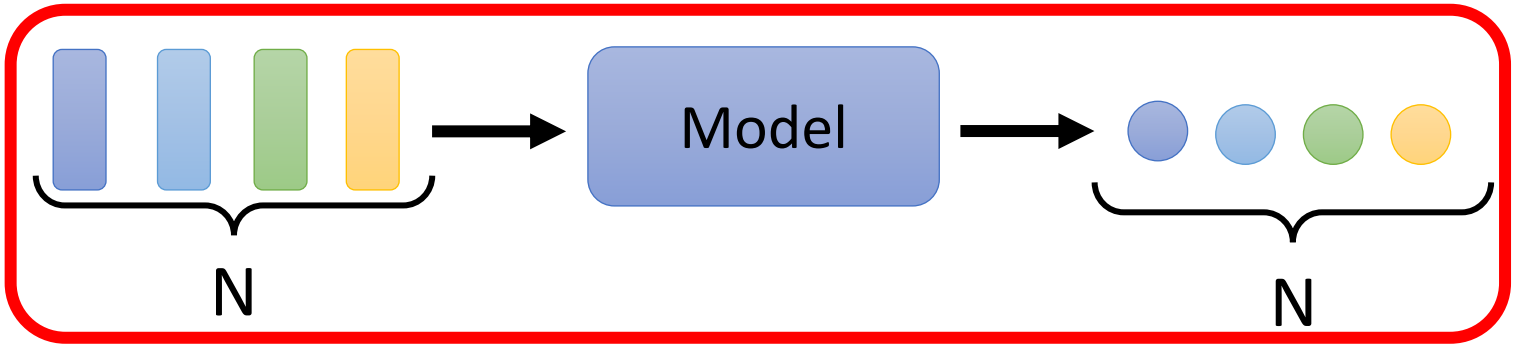
hydrophilicity

給機器一段話，判斷是正面還是負面的

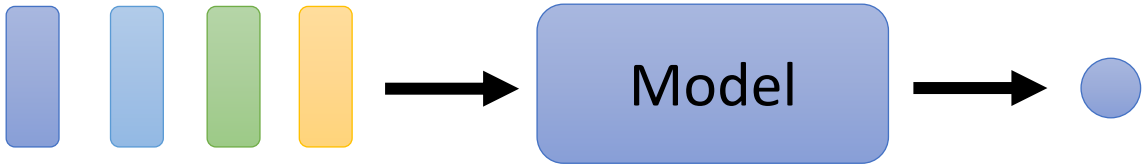
What is the output?

- Each vector has a label.

focus of this lecture



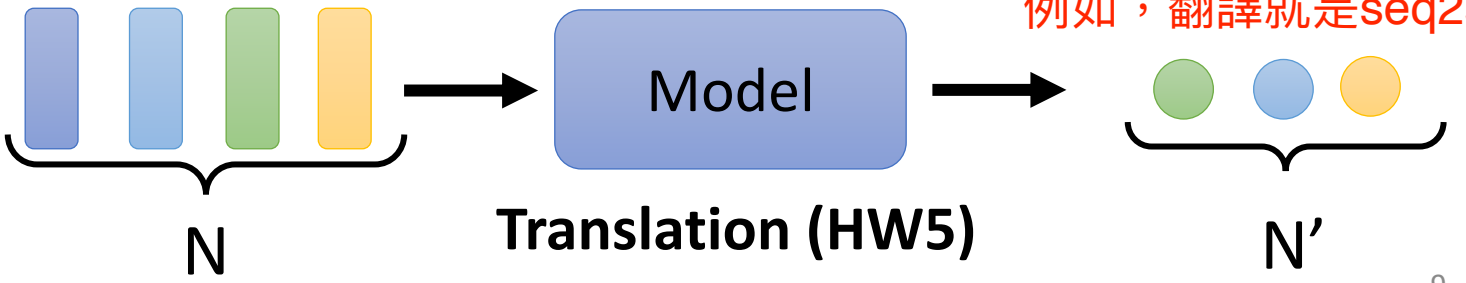
- The whole sequence has a label.



- Model decides the numbers of labels itself.

seq2seq

例如，翻譯就是seq2seq



語音辨識也是，給一段聲音訊號，輸出文字

在作業二，我們也是考慮了前後個五個frame
才來決定這個frame是什麼phonene

Sequence Labeling

假設是詞性標注的問題
就不能單單只把一個vector當作input
必須考慮前後

Is it possible to consider the context?

FC can consider the neighbor

How to consider the whole sequence?

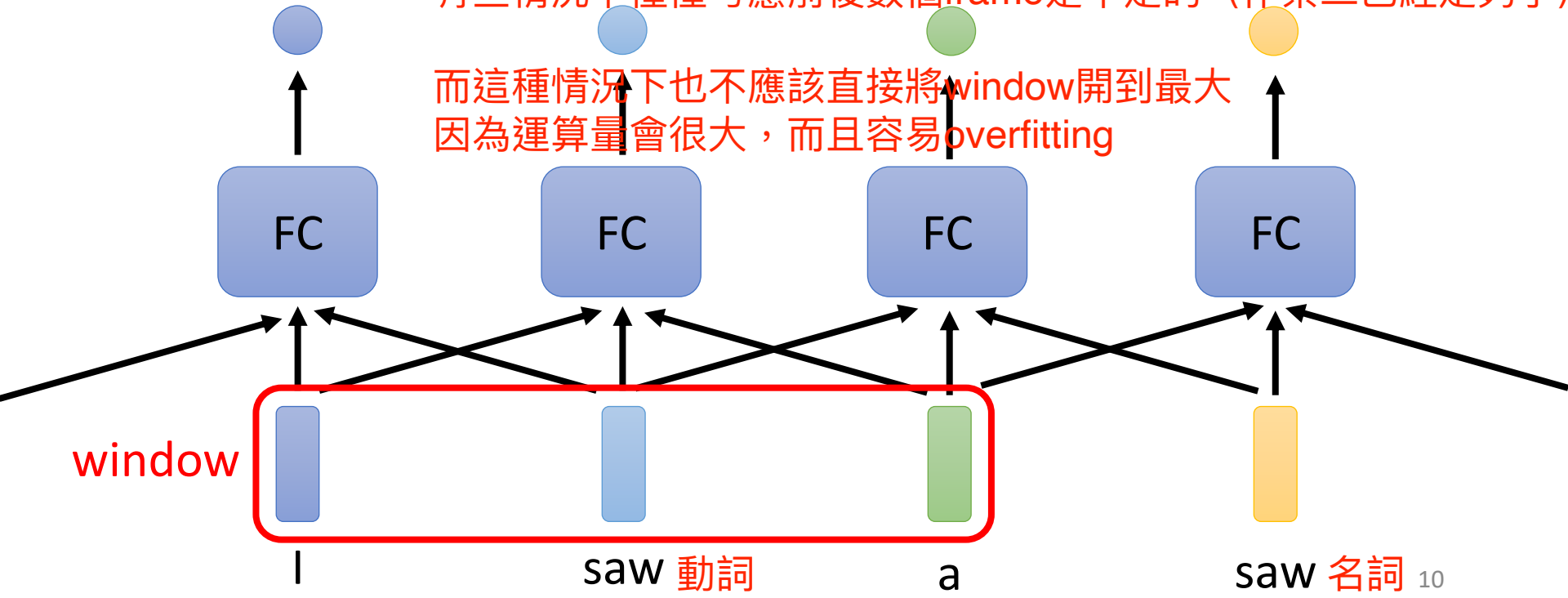
a window covers the whole sequence?

FC

Fully-connected

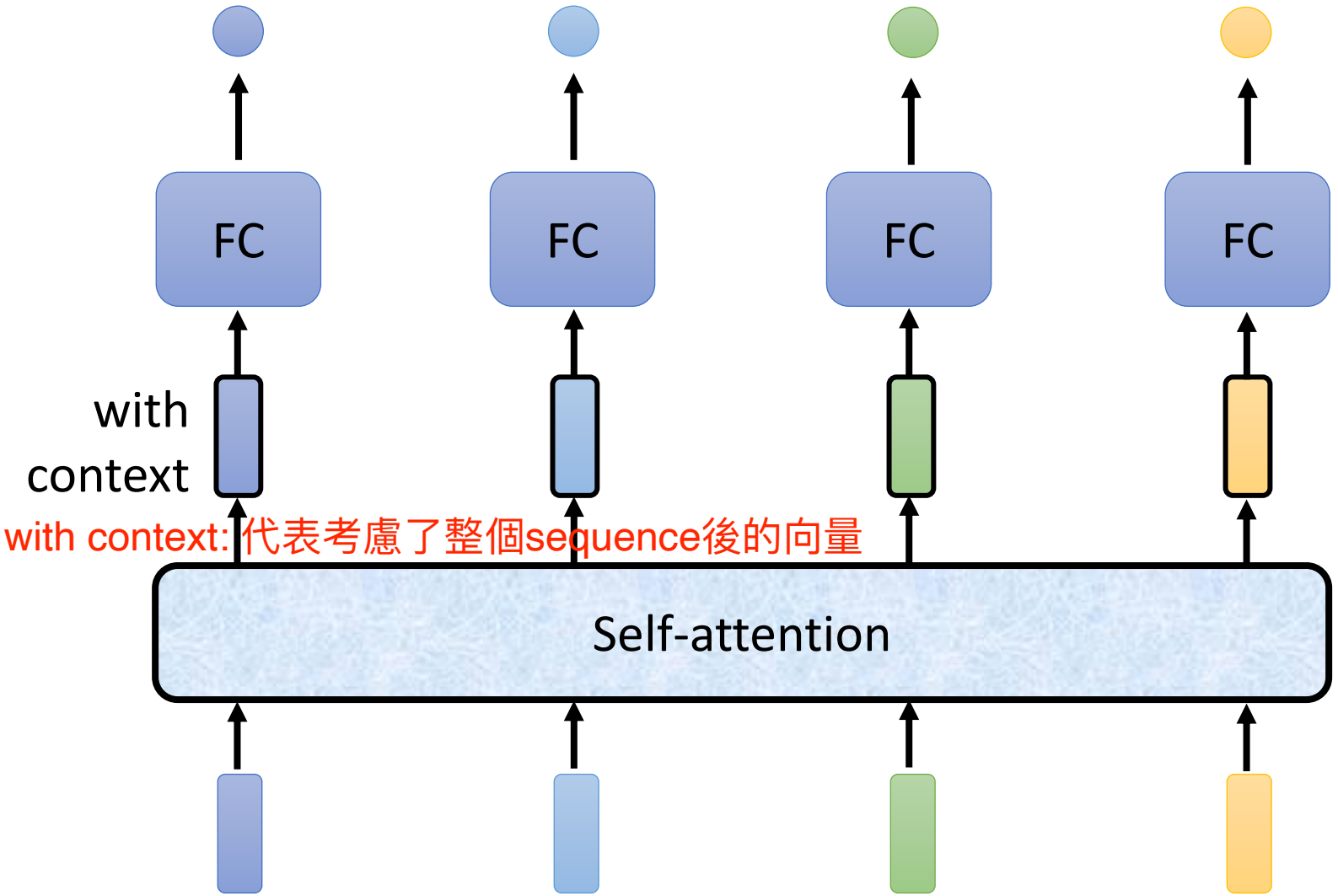
有些情況下僅僅考慮前後數個frame是不足的（作業二已經足夠了）

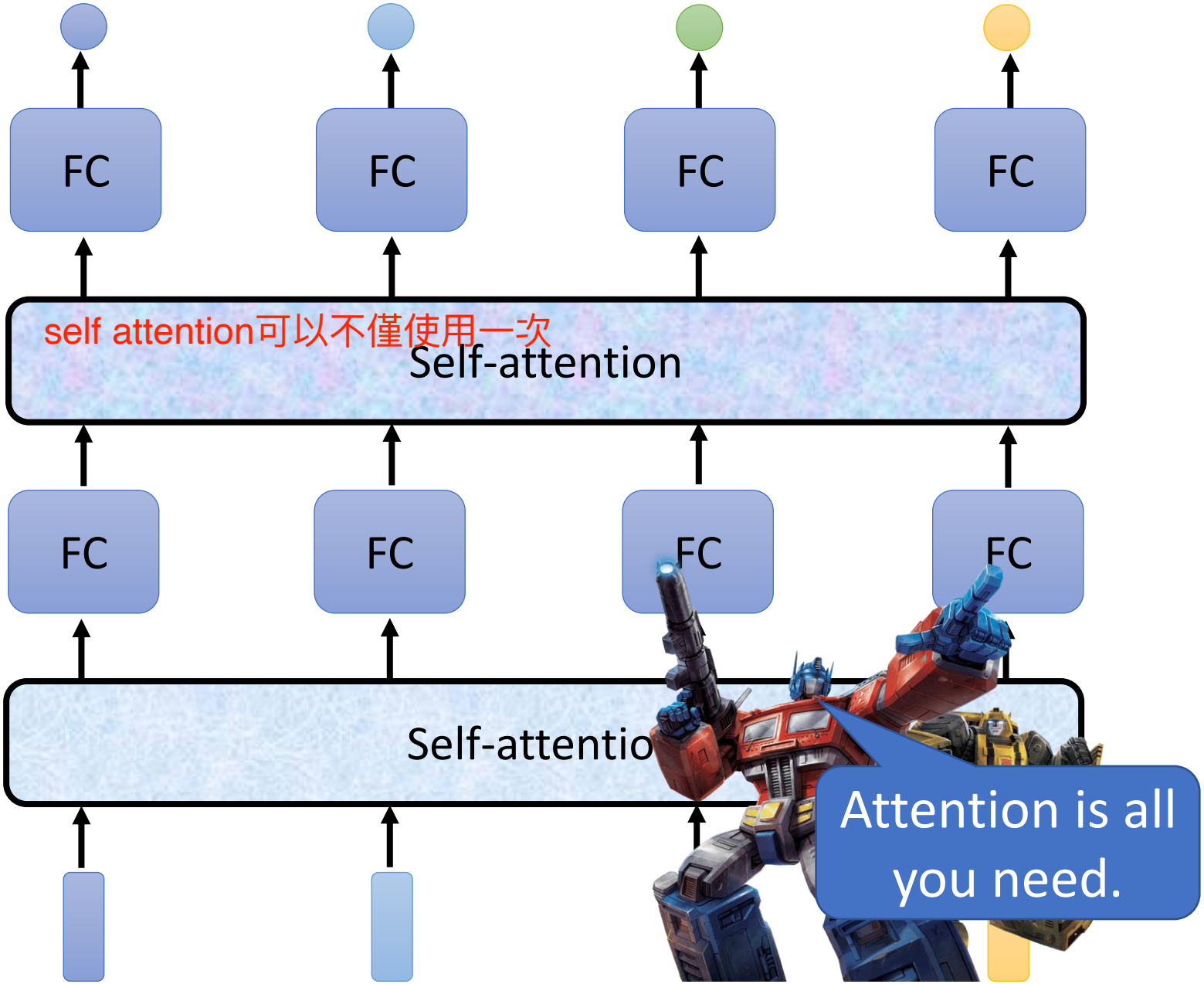
而這種情況下也不應該直接將window開到最大
因為運算量會很大，而且容易overfitting



Self-attention

可以考慮一整個sequence





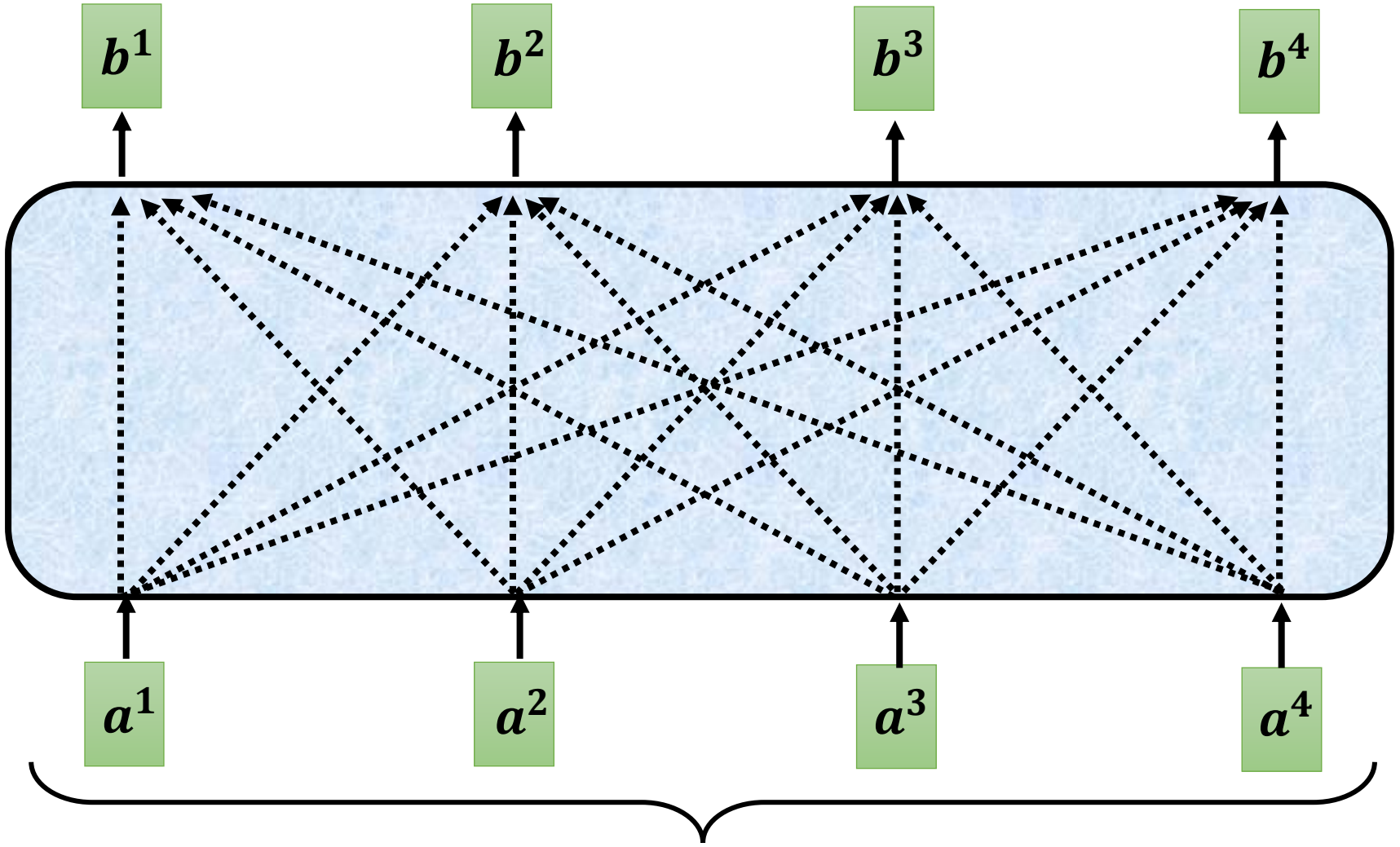
https://arxiv.org/abs/1706.03762

transformer最重要的就是self attention

transformermoduleself-attention

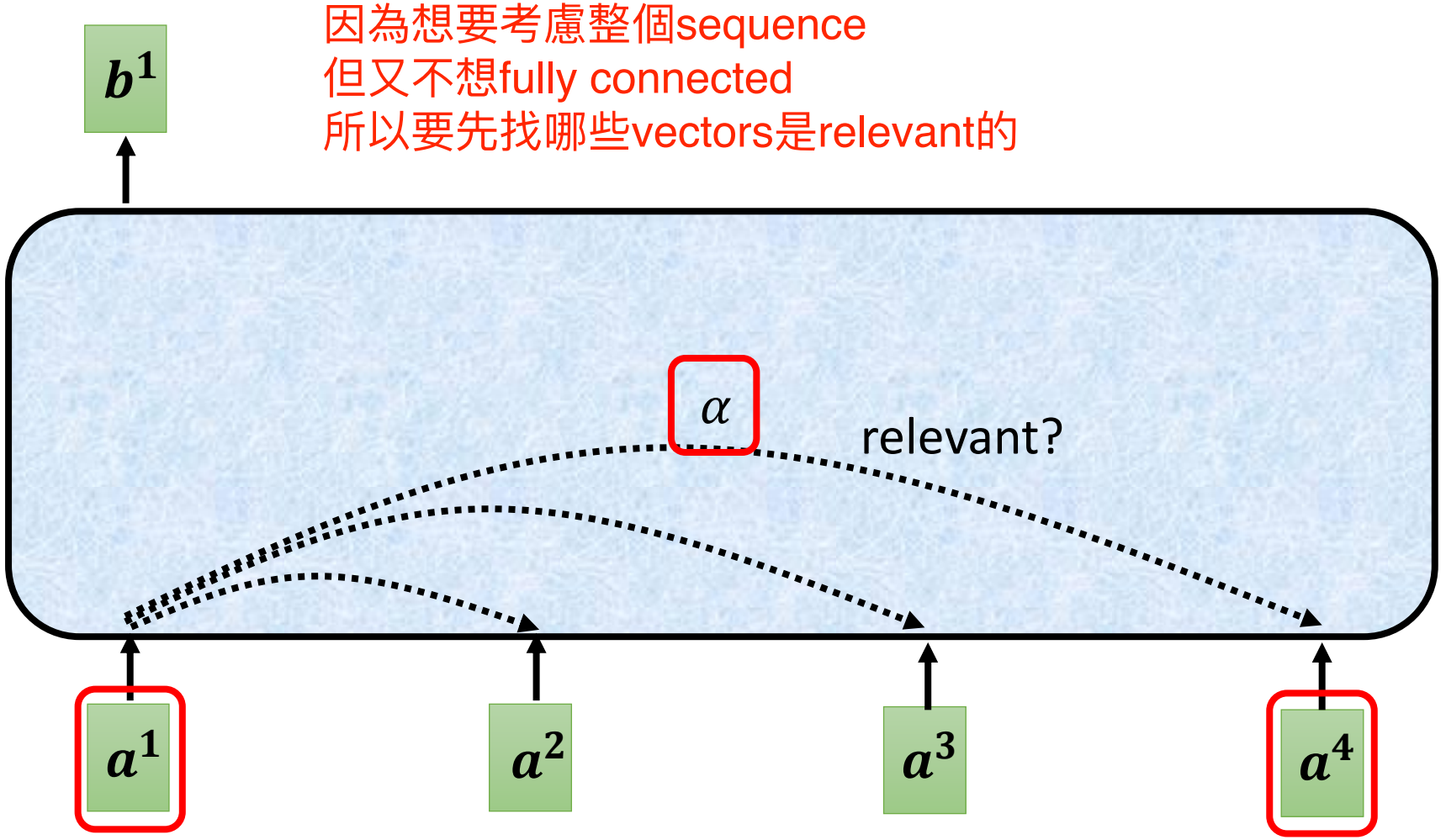
Self-attention

每一個output都考慮了前一層所有的input



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

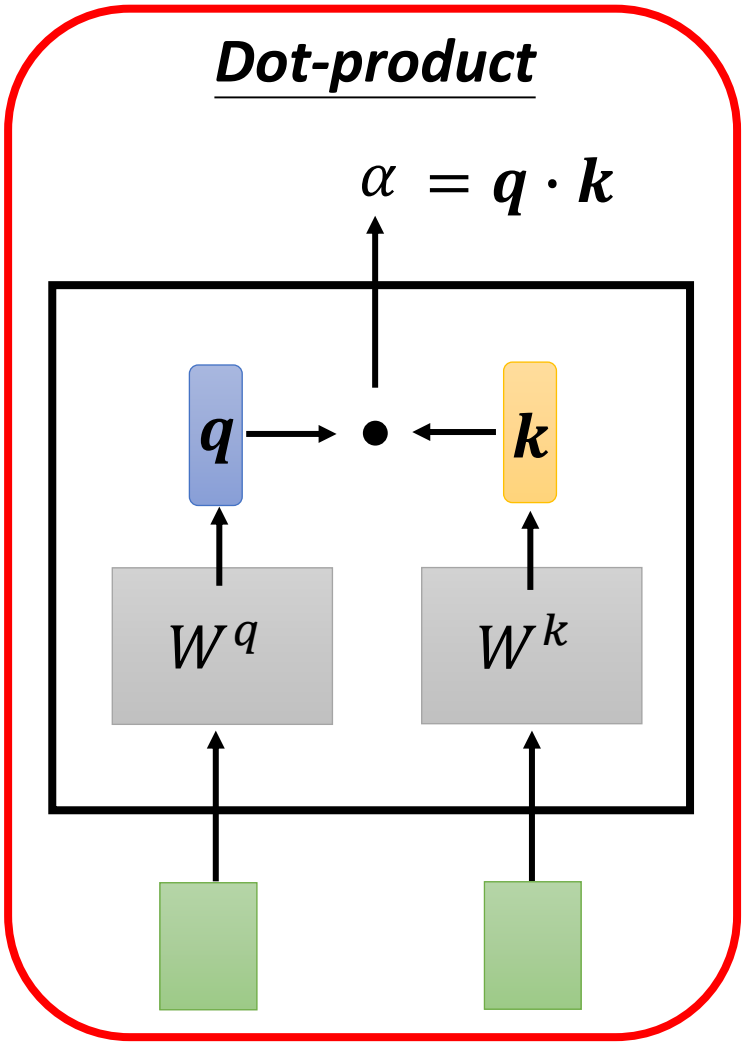
Self-attention

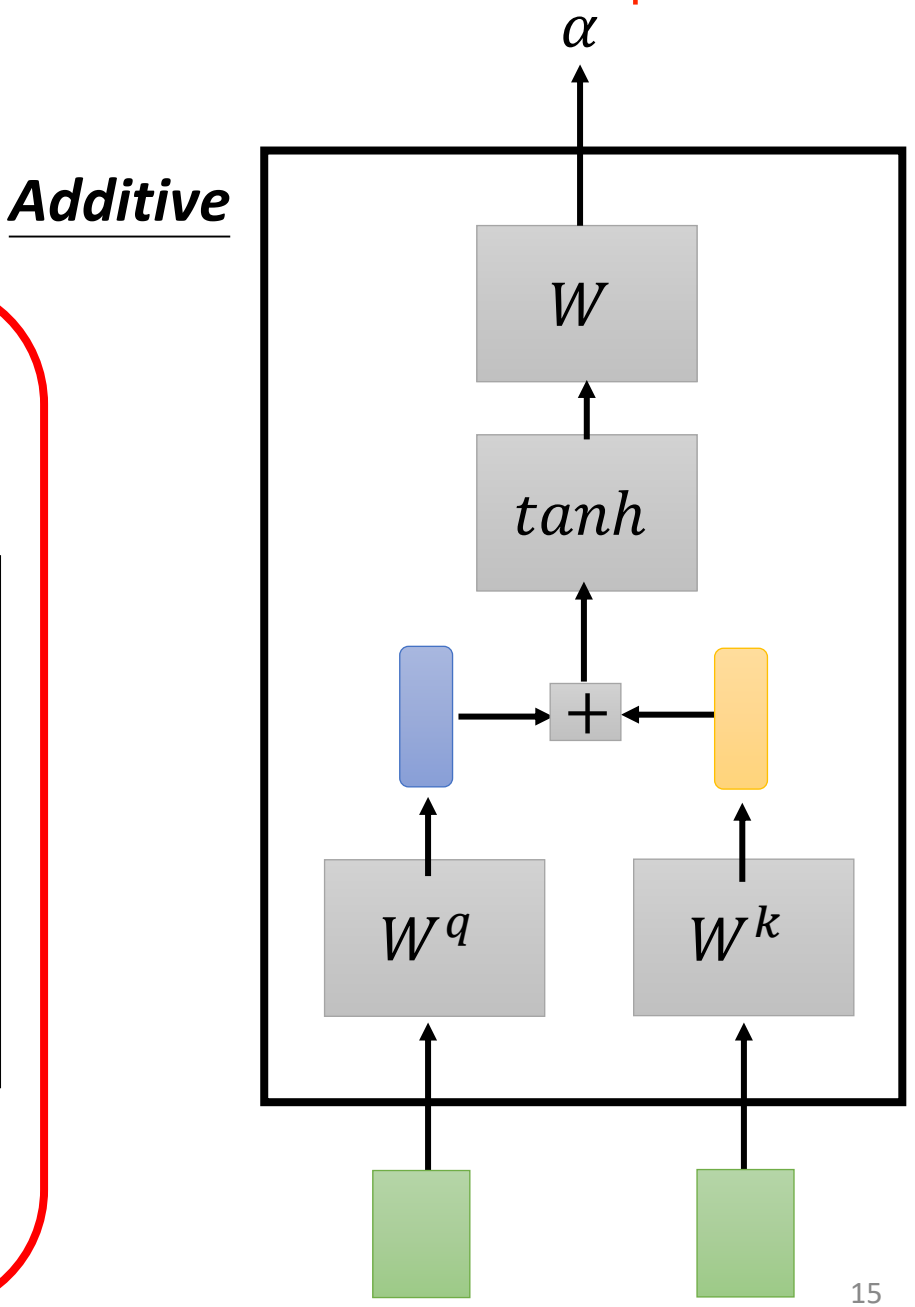


Find the relevant vectors in a sequence

有很多不同計算相關性的方式，在接下來的討論都是使用dot-product

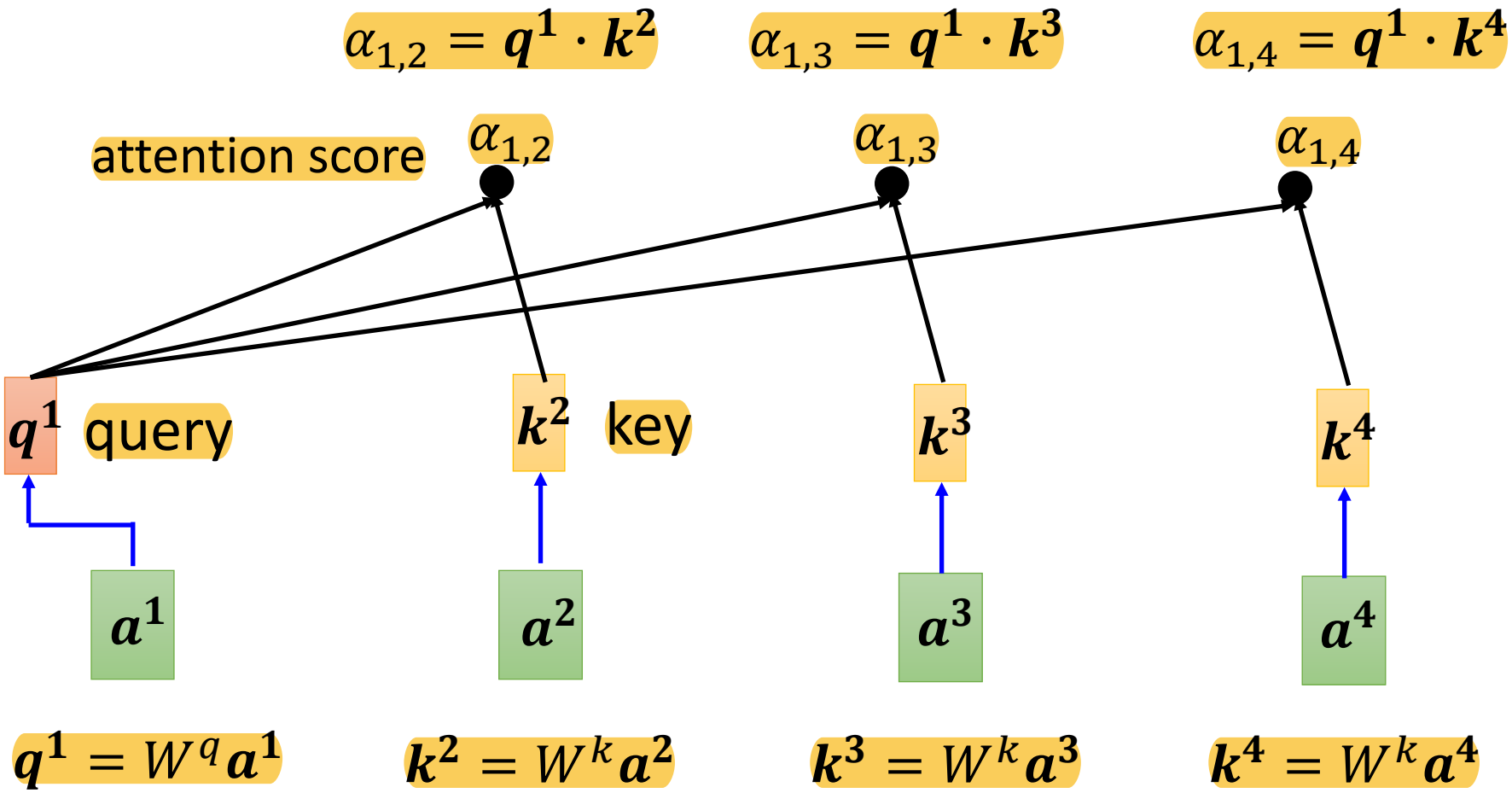
Self-attention





Self-attention

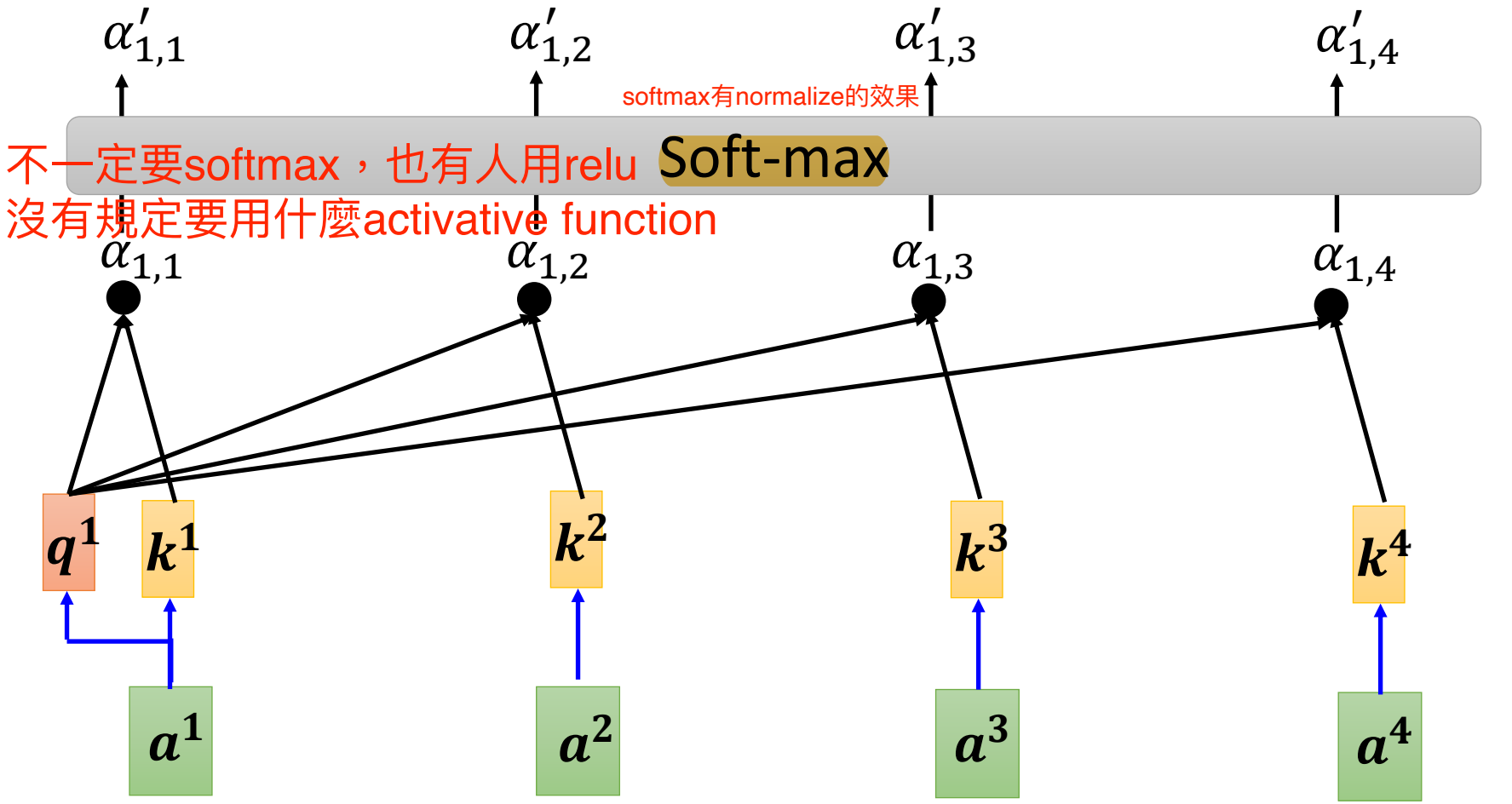
要找與query(a1)相關的key(a2, a3, a4)



Self-attention

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

實作上，也會去計算自己與自己的關聯性



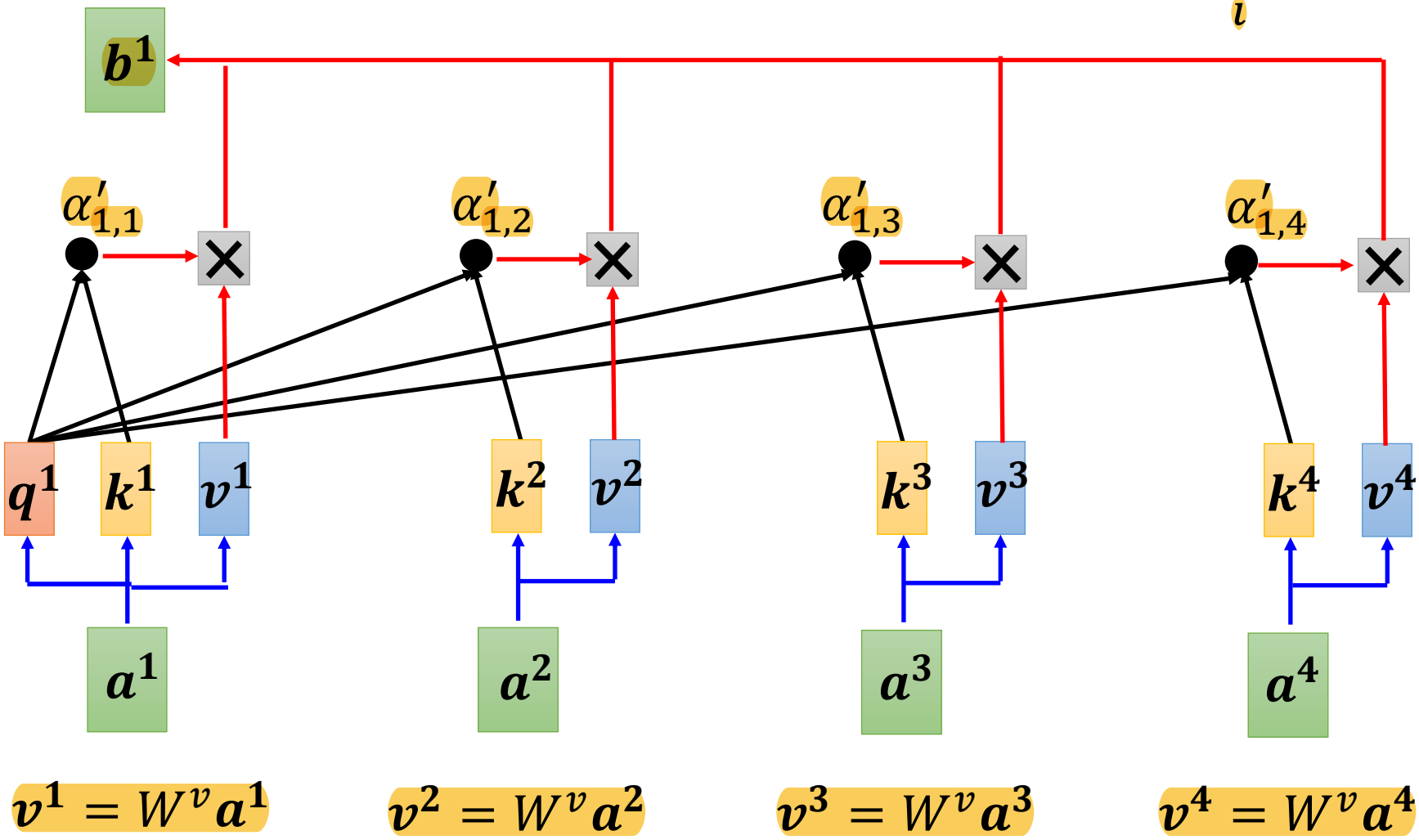
$q^1 = W^q a^1$ $k^2 = W^k a^2$ $k^3 = W^k a^3$ $k^4 = W^k a^4$

$k^1 = W^k a^1$

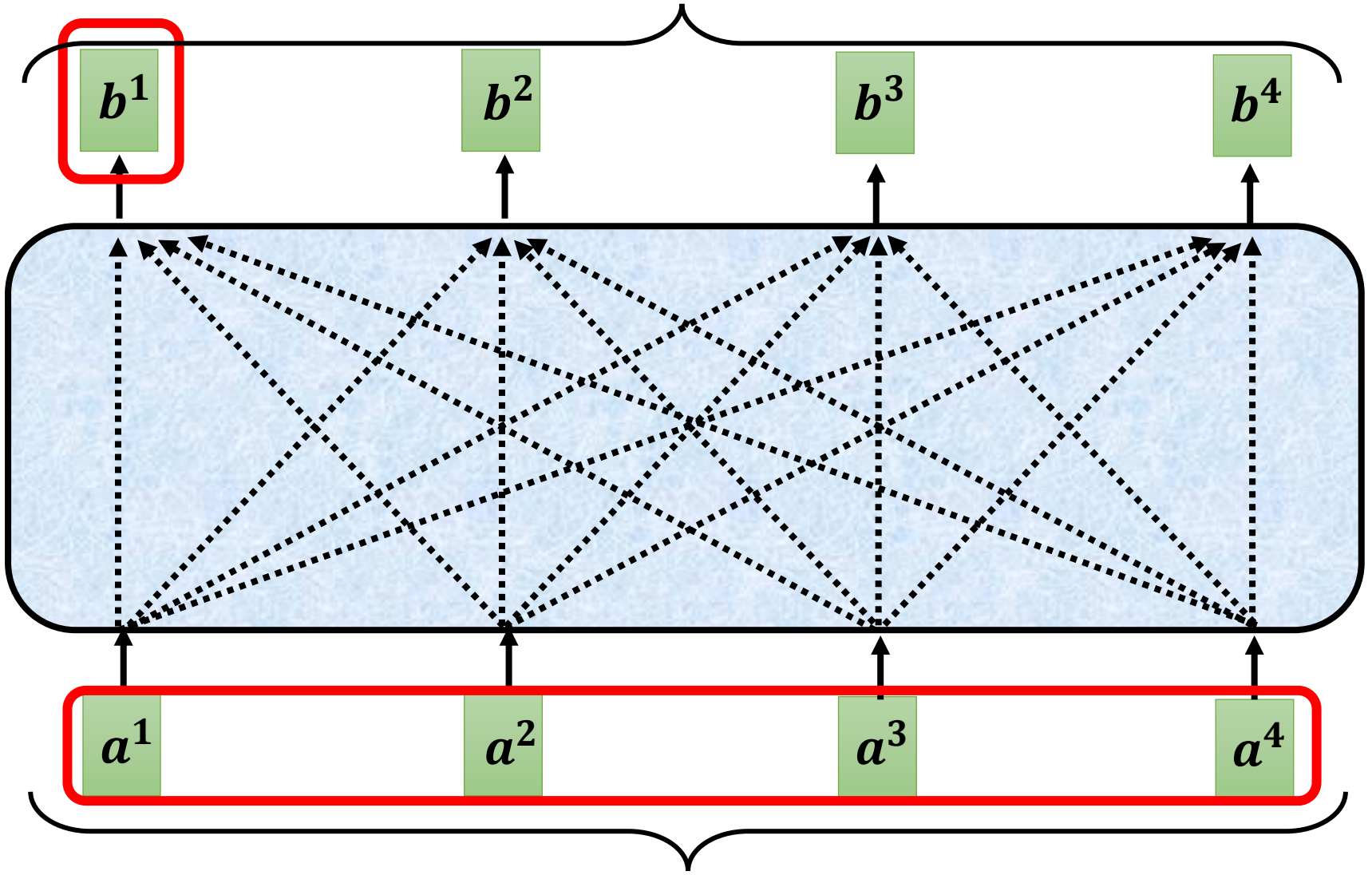
Self-attention

Extract information based on attention scores

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



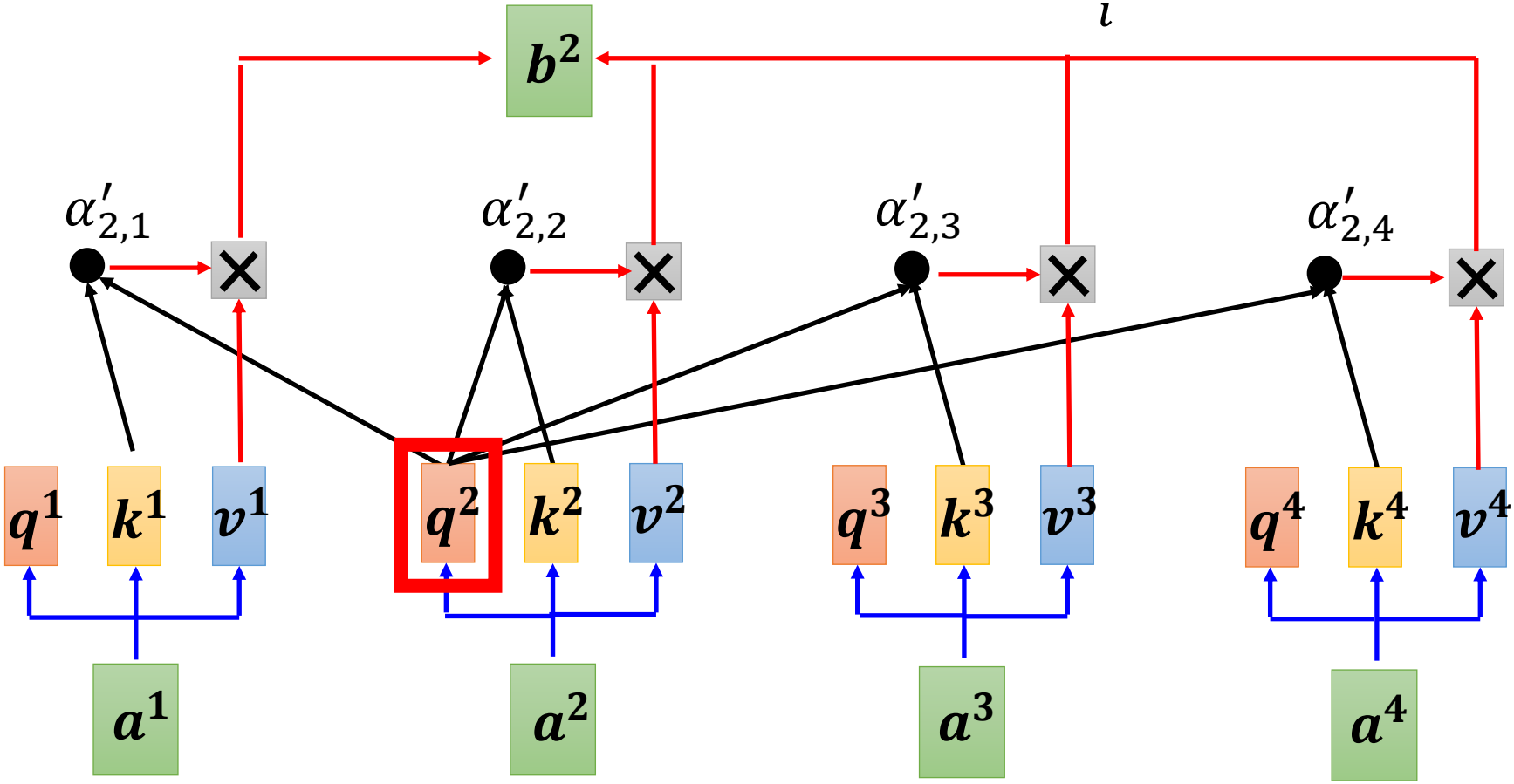
Self-attention



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

Self-attention

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



Self-attention

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

$$q^1\ q^2\ q^3\ q^4 = W^q\ a^1\ a^2\ a^3\ a^4$$

Q $\quad$ I

$$k^i = W^k a^i$$

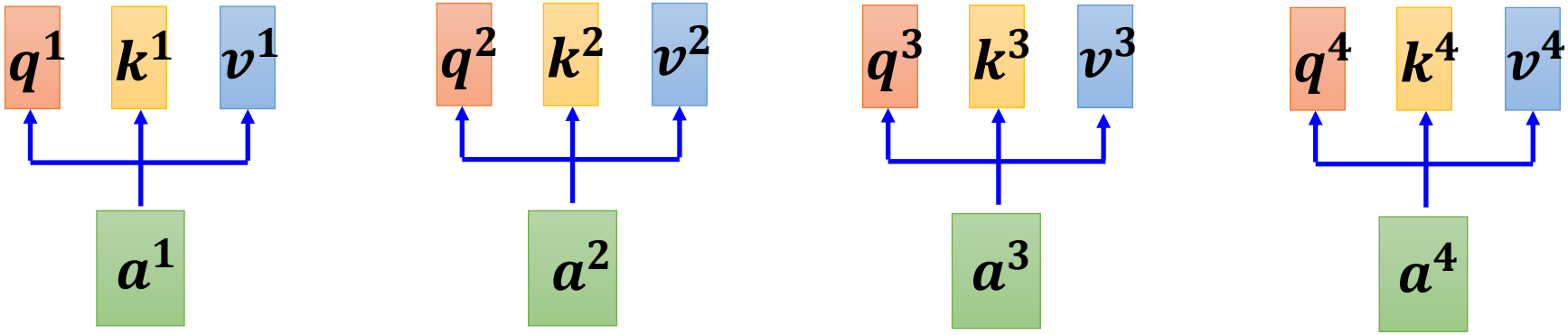
$$k^1\ k^2\ k^3\ k^4 = W^k\ a^1\ a^2\ a^3\ a^4$$

K $\quad$ I

$$v^i = W^v a^i$$

$$v^1\ v^2\ v^3\ v^4 = W^v\ a^1\ a^2\ a^3\ a^4$$

V $\quad$ I

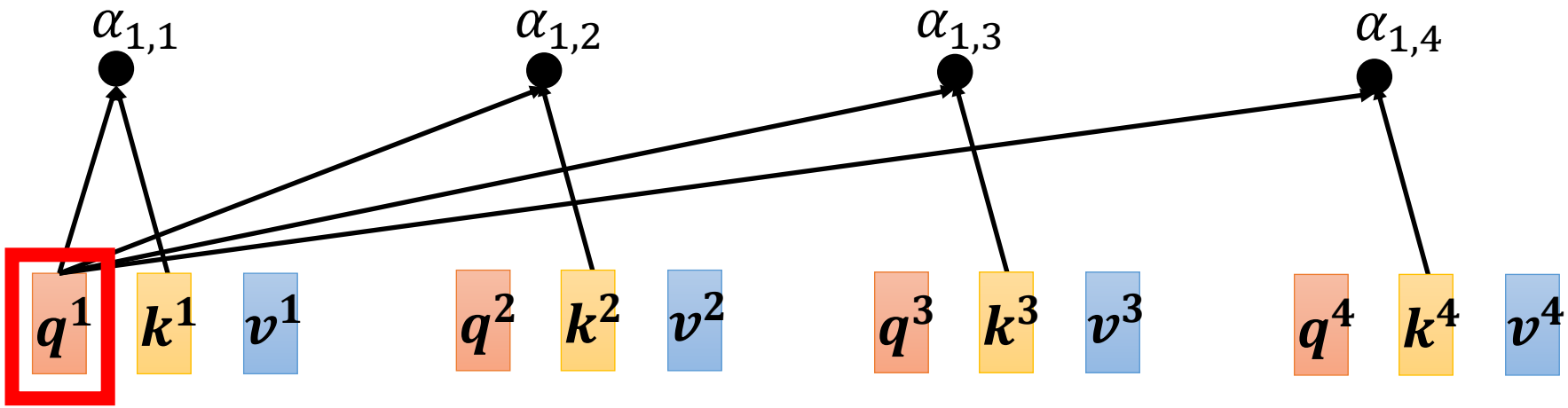


Self-attention

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

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

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

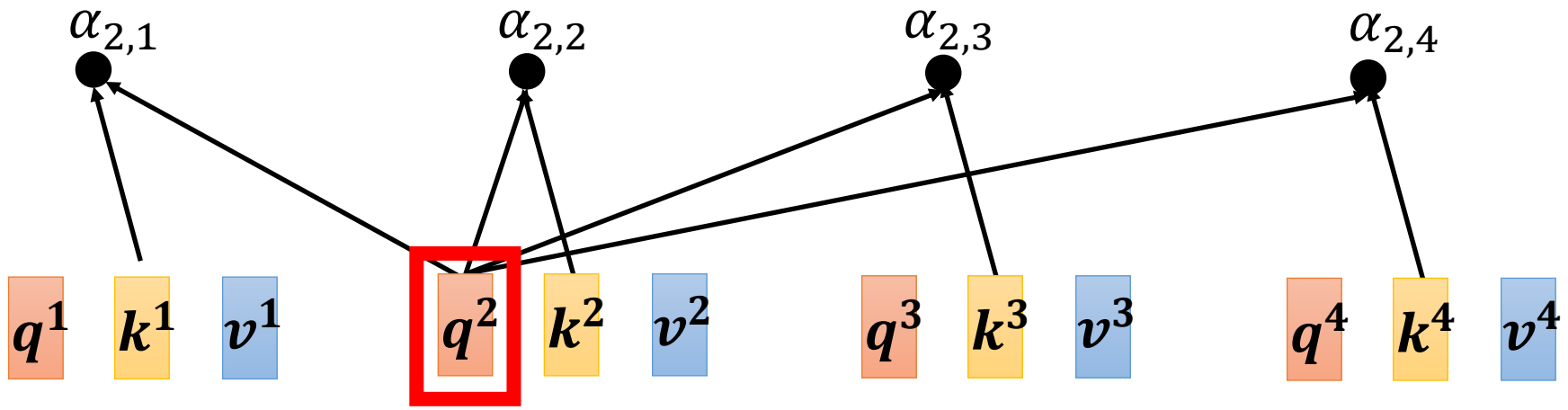


Self-attention

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

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

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



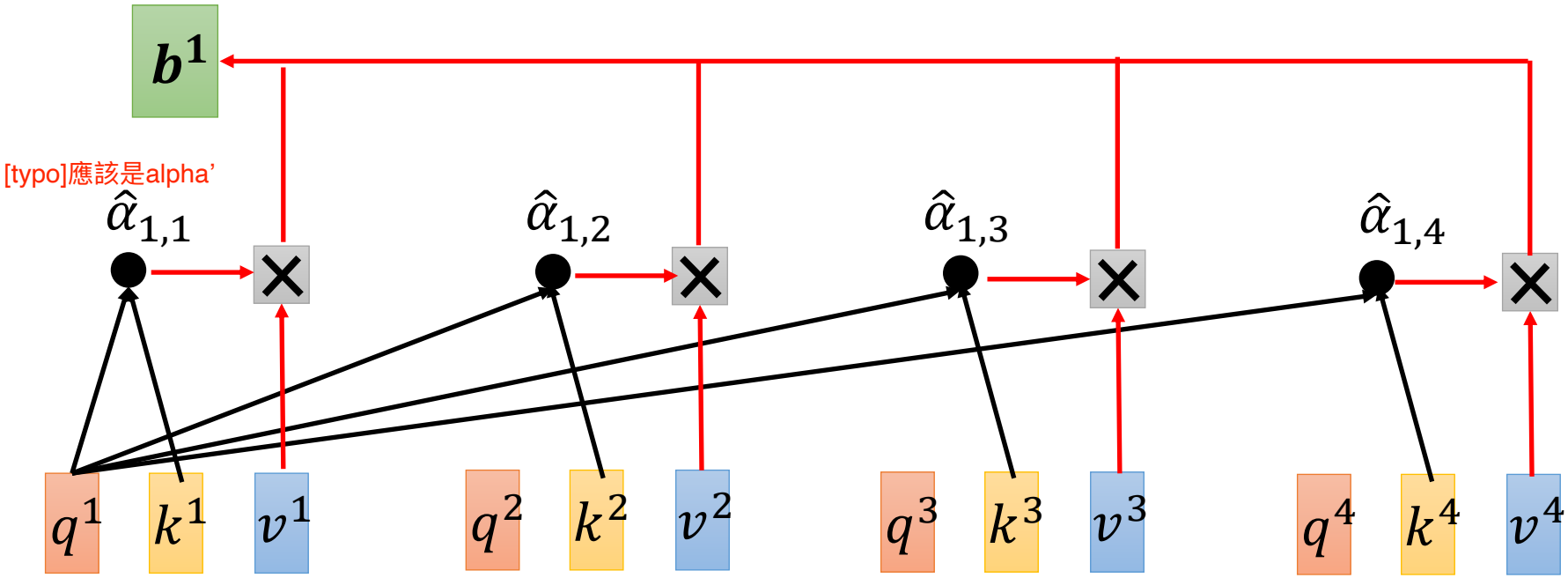
$$\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix} = A'$$

使用softmax或是其他normalization

$$\begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1} \\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2} \\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3} \\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4} \end{bmatrix} = A$$

$$\begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix} = K^T \qquad \begin{bmatrix} q^1 & q^2 & q^3 & q^4 \end{bmatrix} = Q$$

Self-attention



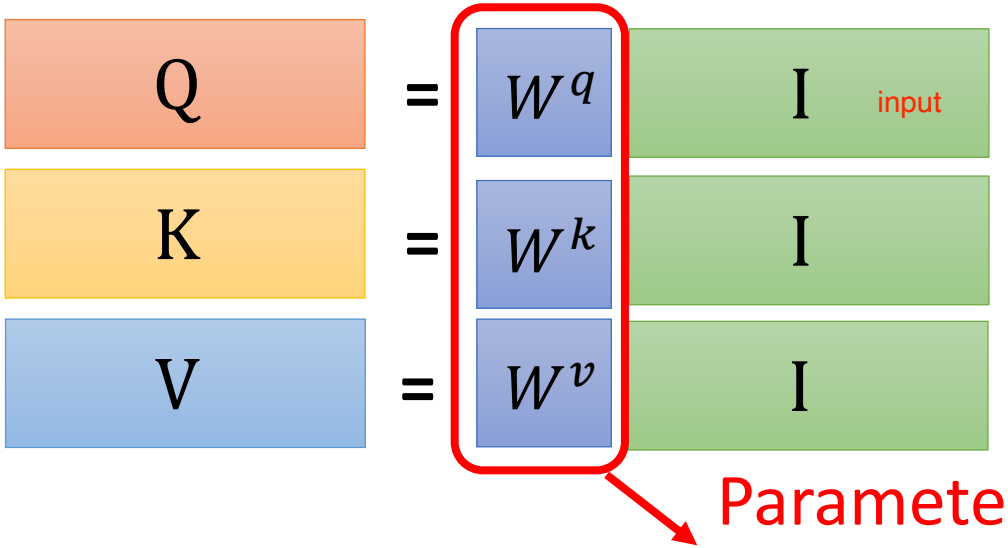
$b^1\ b^2\ b^3\ b^4$ = $v^1\ v^2\ v^3\ v^4$

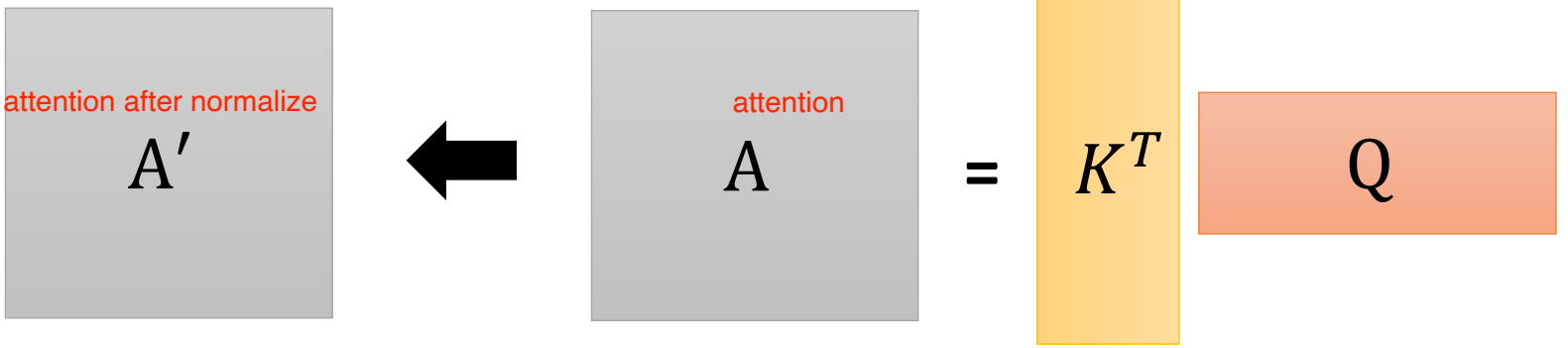
O V

$$\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}$$

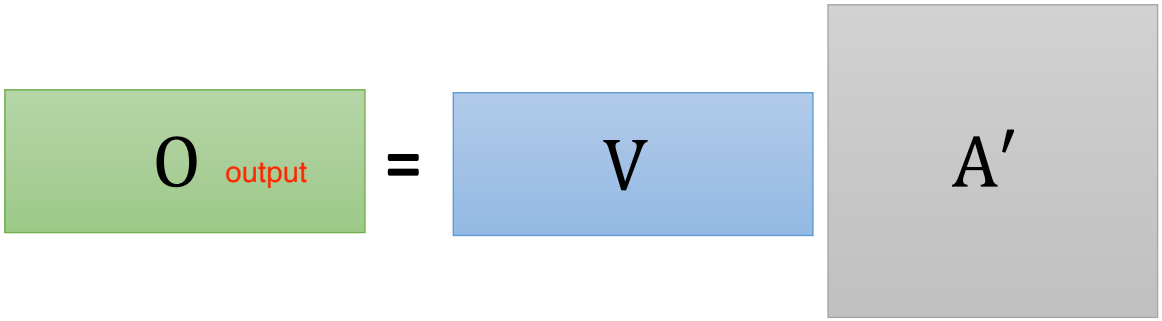
A'

Self-attention





Attention Matrix



Multi-head Self-attention

Different types of relevance

可能有很多種相關性，因此只有一組q, k, v可能會不夠好

作業四用比較少的head就可以過medium base line了

但並不是所有的task都是用比較少的head比較好

例如翻譯和語音辨識

但需要多少head是hyper parameter

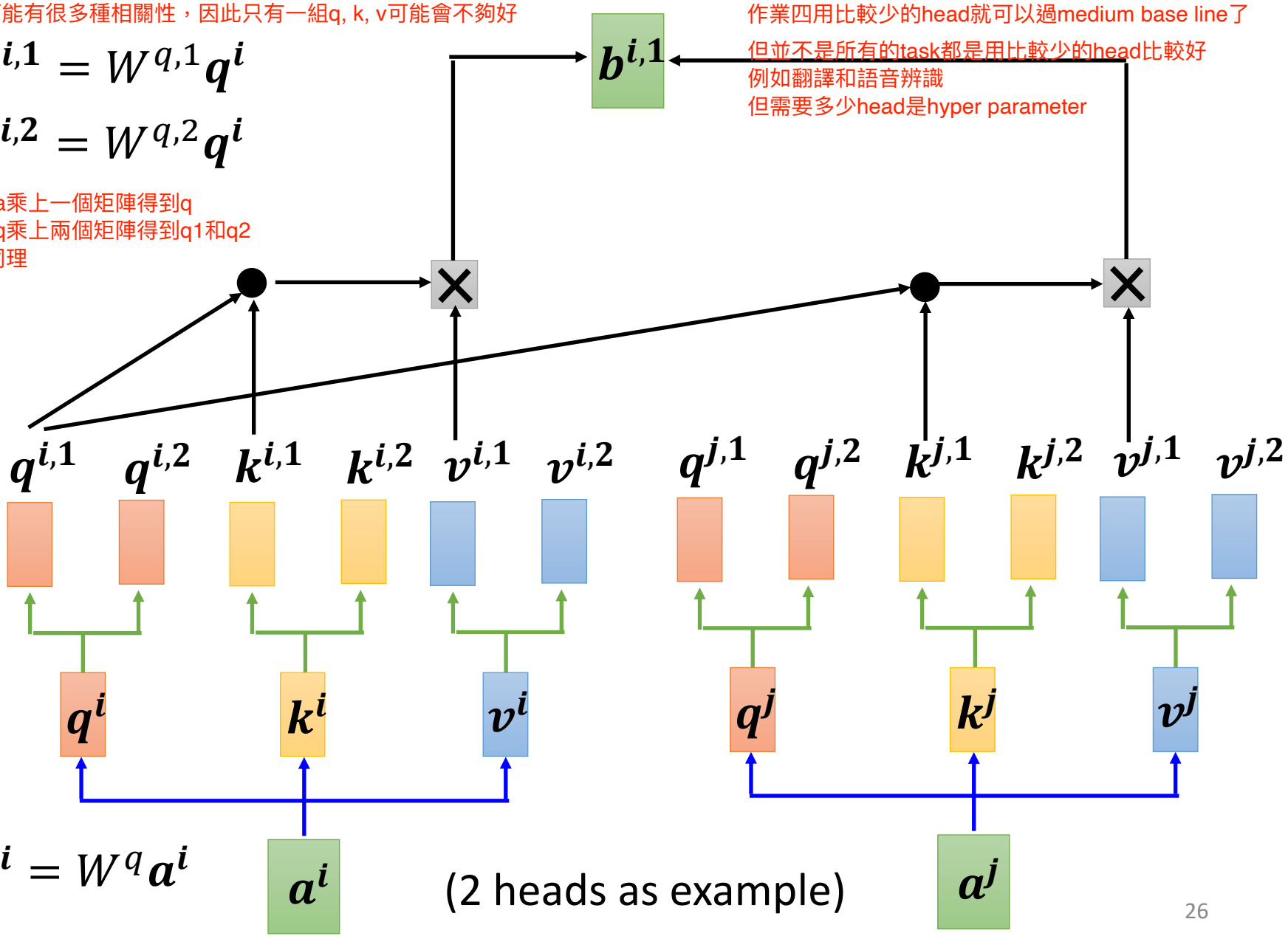
$$q^{i,1} = W^{q,1} q^i$$

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

先將a乘上一個矩陣得到q

再將q乘上兩個矩陣得到q1和q2

k, v同理



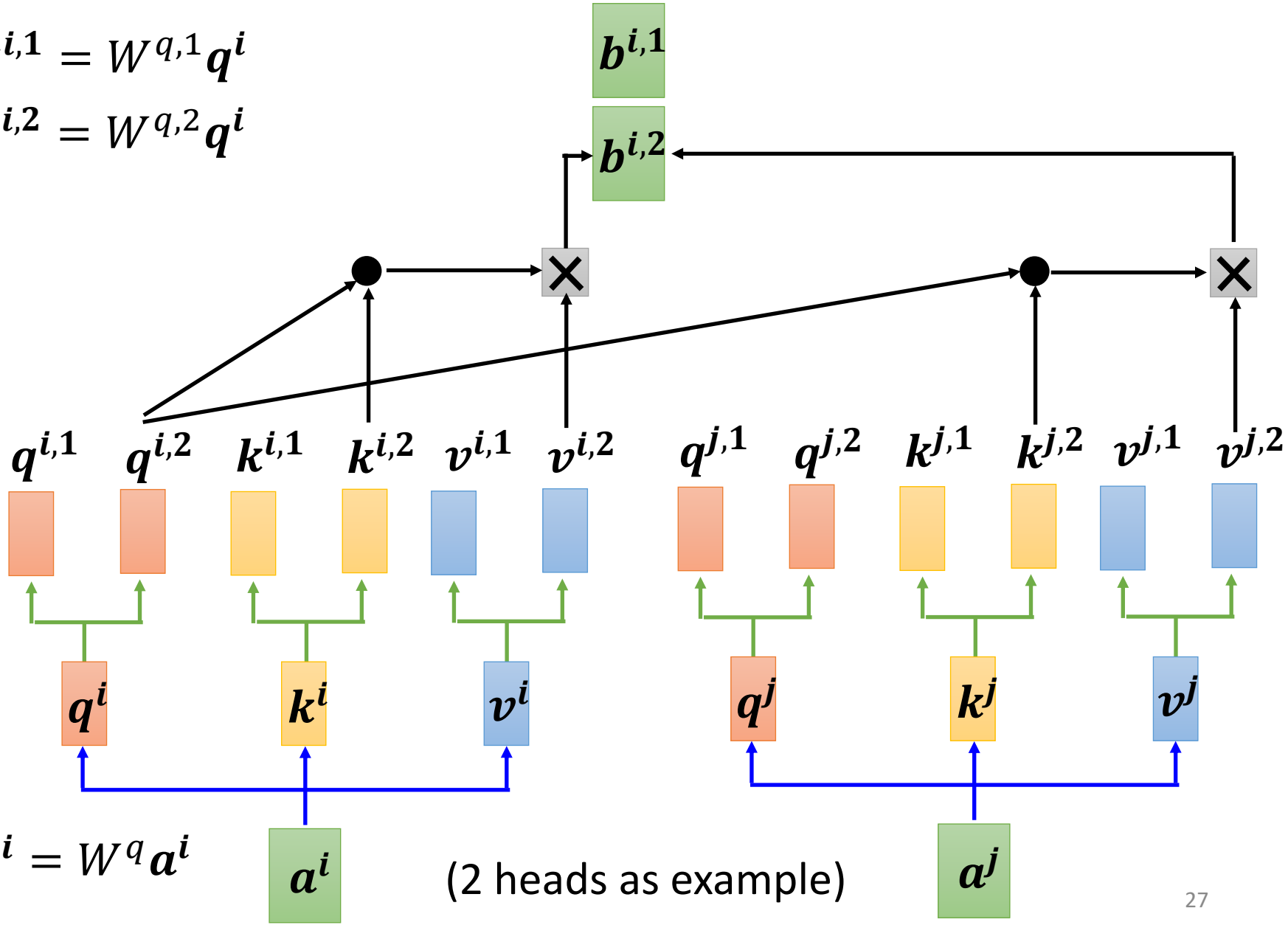
$$q^i = W^q a^i$$

(2 heads as example)

Multi-head Self-attention Different types of relevance

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

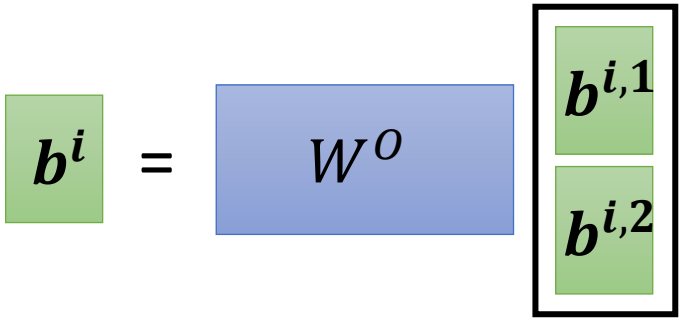
$$q^{i,2} = W^{q,2} q^i$$

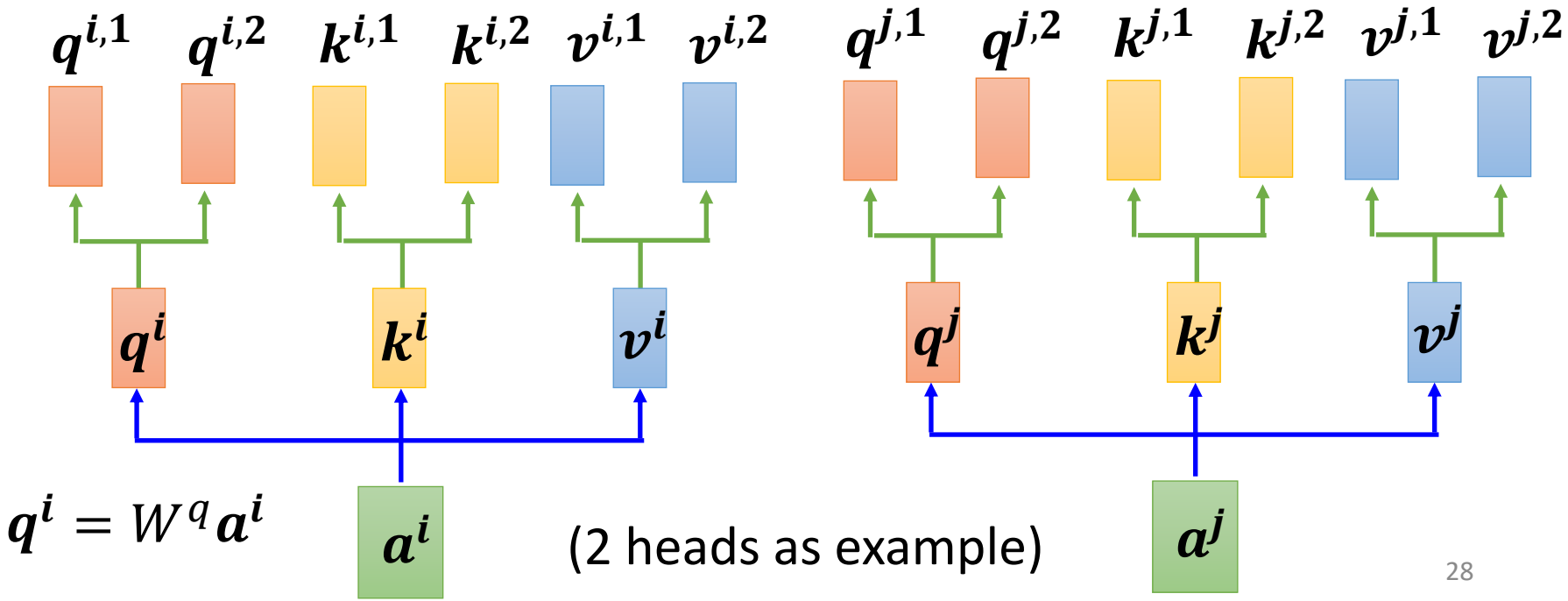


$$q^i = W^q a^i$$

(2 heads as example)

Multi-head Self-attention Different types of relevance





Positional Encoding

做self-attention時若覺得position的資訊是重要的
可以使用position encoding

在attention is all you need這篇paper中
使用這個matrix來做positional encoding
其中第i個column代表第i個位置的positional vector

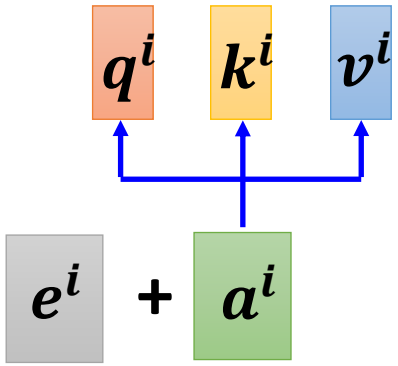
Each column represents a positional vector e^i

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

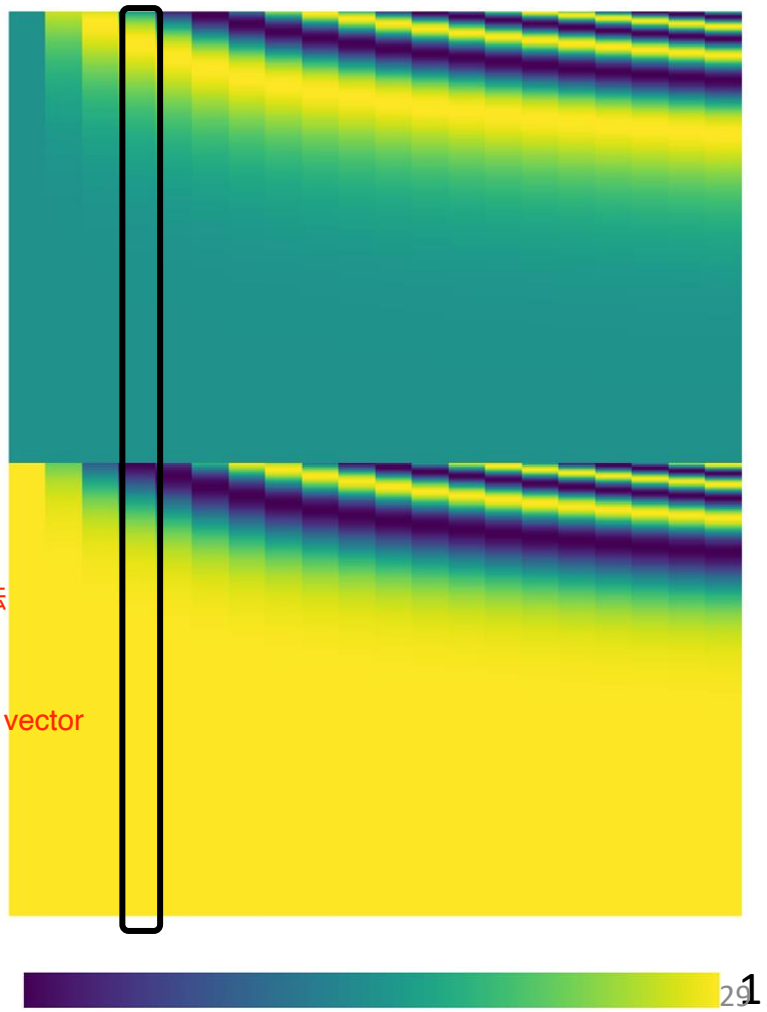
可是手刻的position vector會有問題
就是當今天sequence長度改變了話
要怎麼辦？

但在attention is all you need這個paper當中

他有一種自動生成
position vector的演算法



至於怎麼樣產生position vector
是個尚待研究的問題



自己製造一個e_i
代表每個不同position的位置向量
接著把它加上a_i就可以包括到位置的資訊了

有許多論文提出各種不同的position encoding

https://arxiv.org/abs/2003.09229

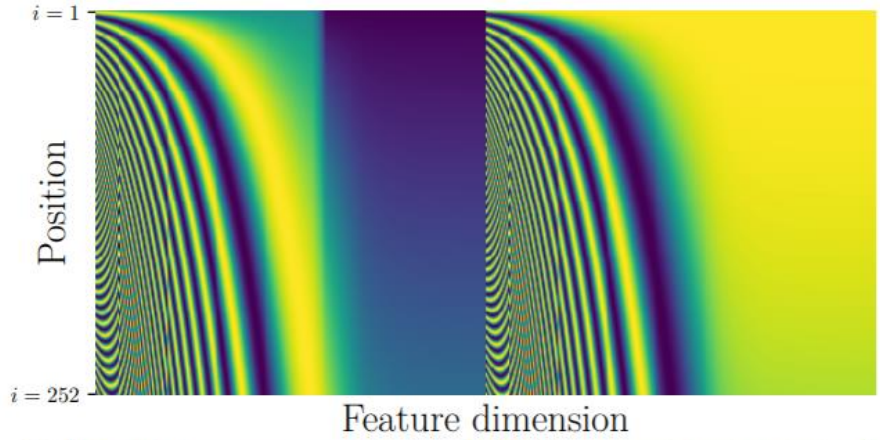
每個row是一個position vector

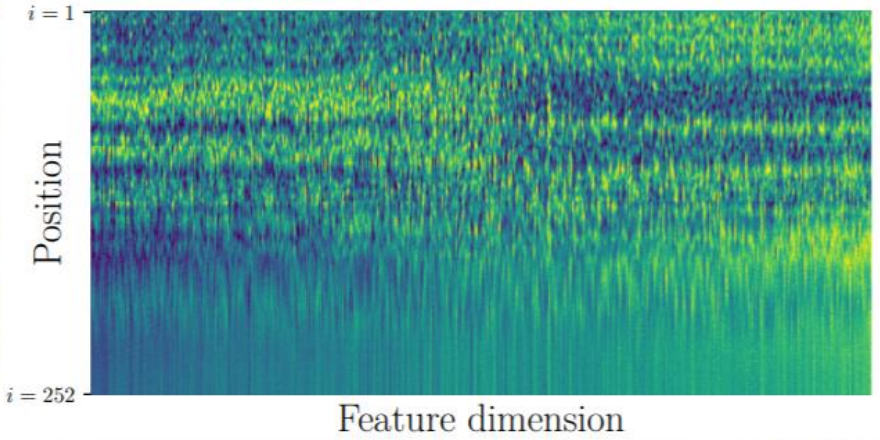
Table 1. Comparing position representation methods

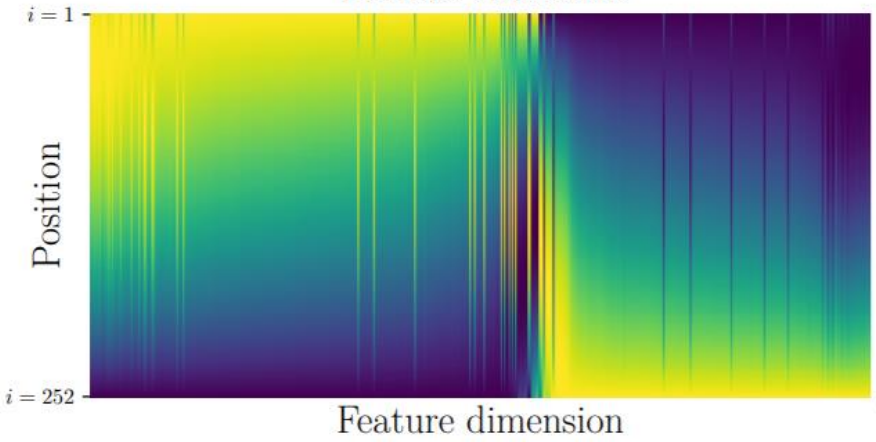
Methods	Inductive	Data-Driven	Parameter Efficient
Sinusoidal (Vaswani et al., 2017)	✓	✗	✓
Embedding (Devlin et al., 2018)	✗	✓	✗
Relative (Shaw et al., 2018)	✗	✓	✓
This paper	✓	✓	✓

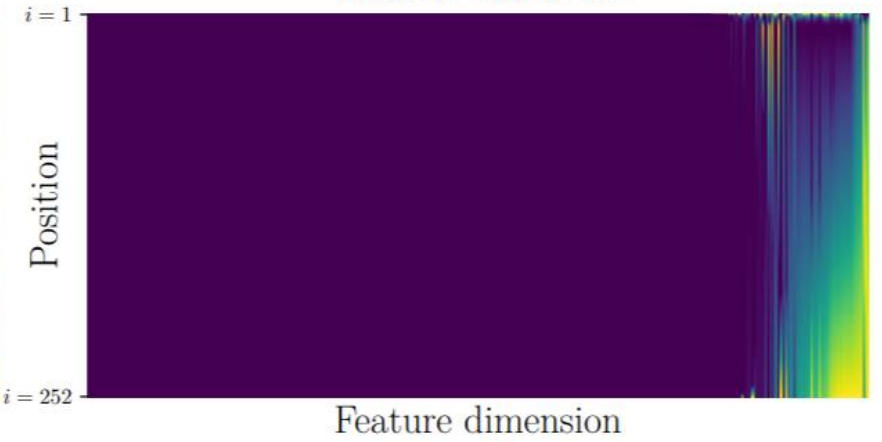
(a) Sinusoidal

(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)!

至於範圍要多大，就要看今天要做的task是什麼
舉例來說，如果是要判斷某一個frame是屬於什麼phomene
不需要聽完整段演講，只需要前後多看幾個frame就好

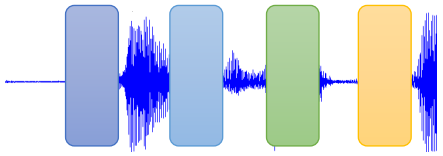
https://arxiv.org/abs/1910.12977

Self-attention for Speech

每個frame是10ms，1sec的聲音訊號就有100個frame了

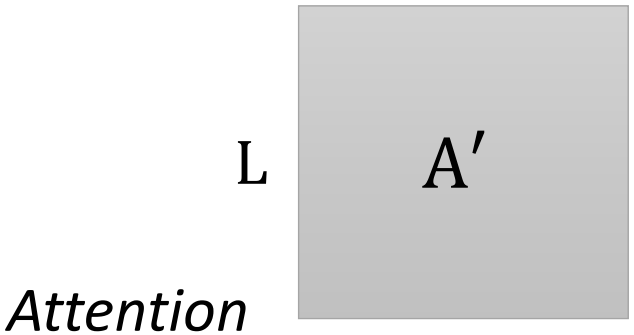
Speech is a very long vector sequence.

10ms



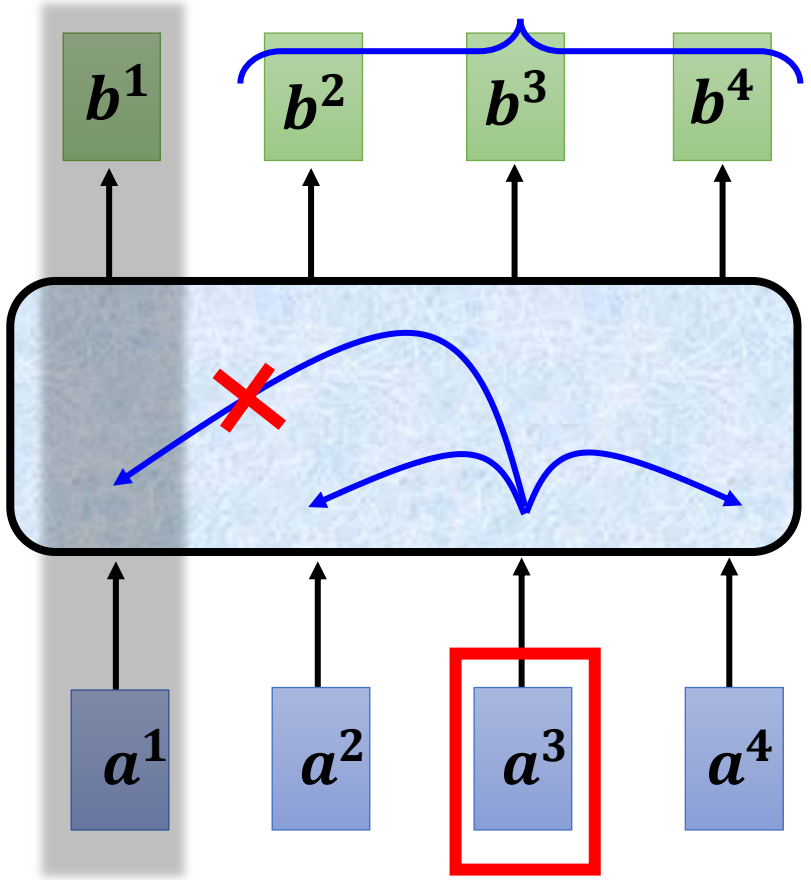
If input sequence is length L

計算attention matrix的複雜度是frame數量的平方



Attention Matrix

Attention in a range

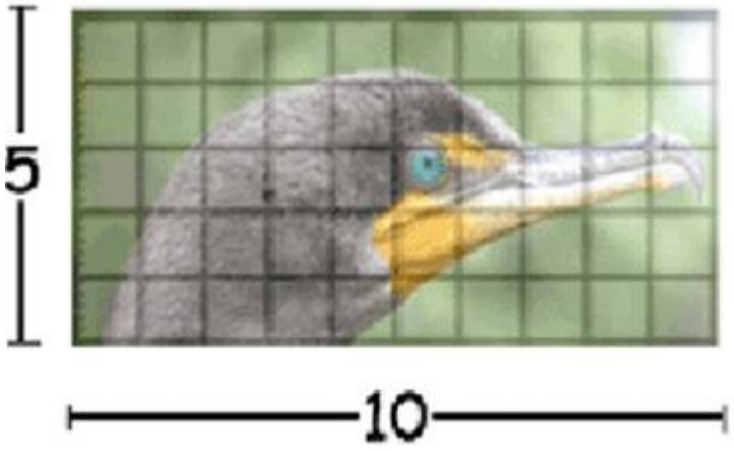


Truncated Self-attention

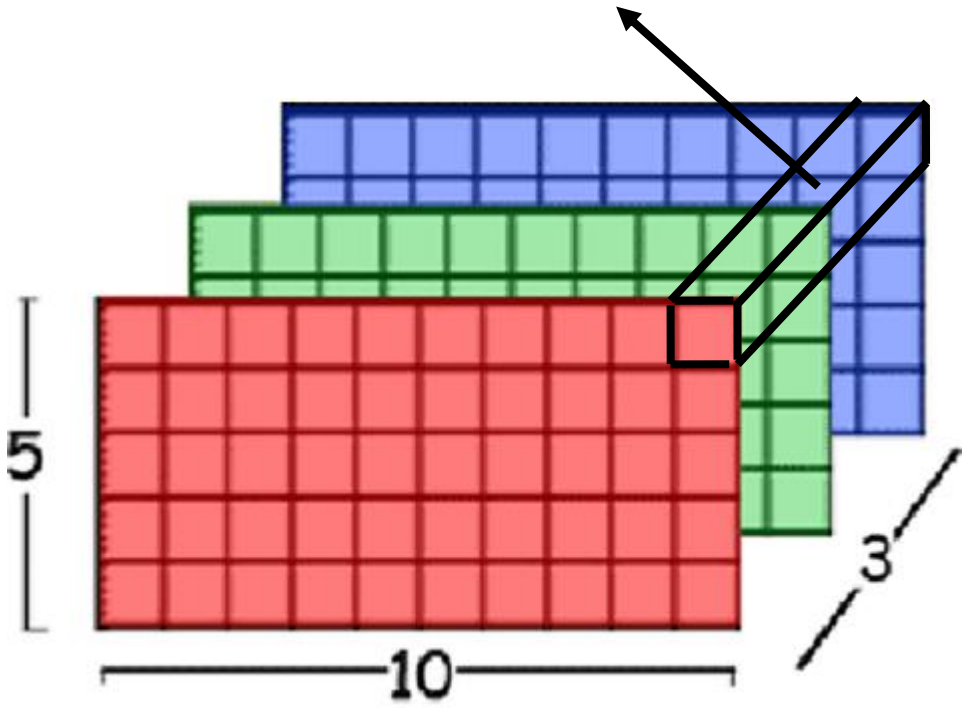
只看某個範圍內的值就好

Self-attention for Image

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

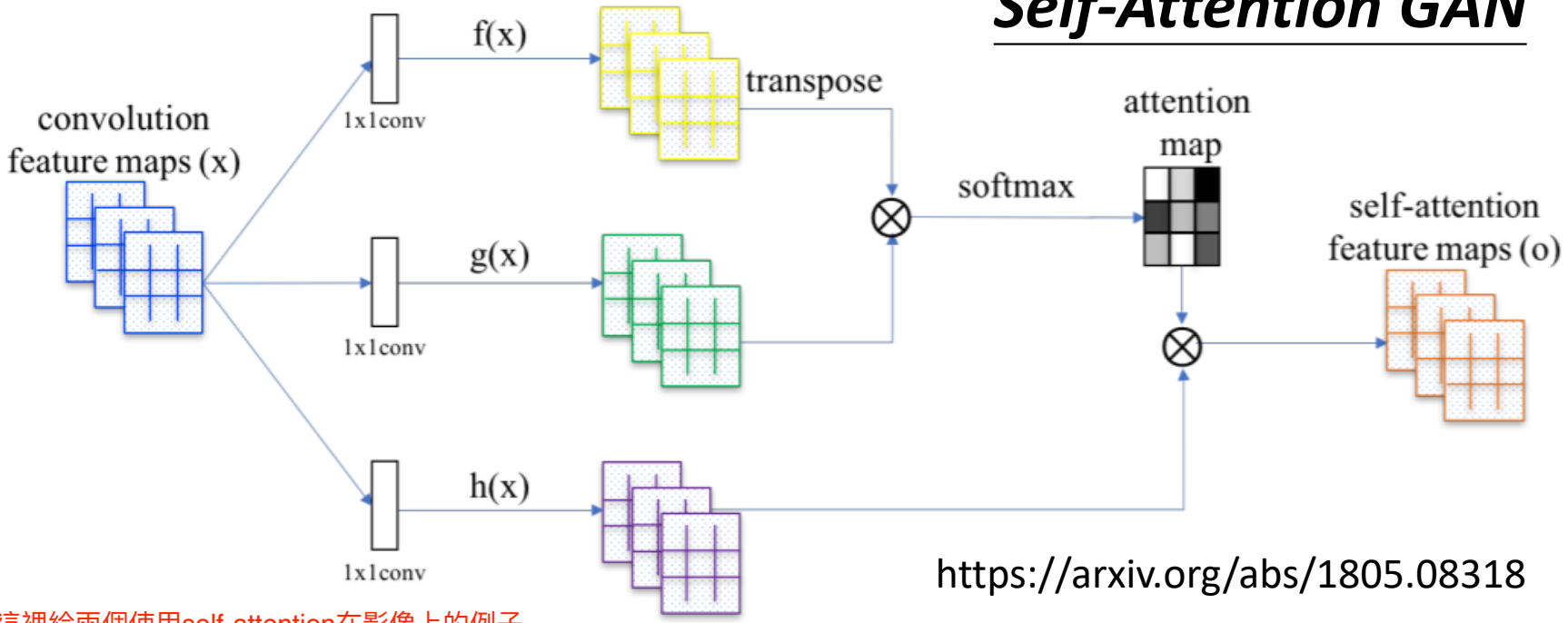


This is a vector.



Source of image: https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184

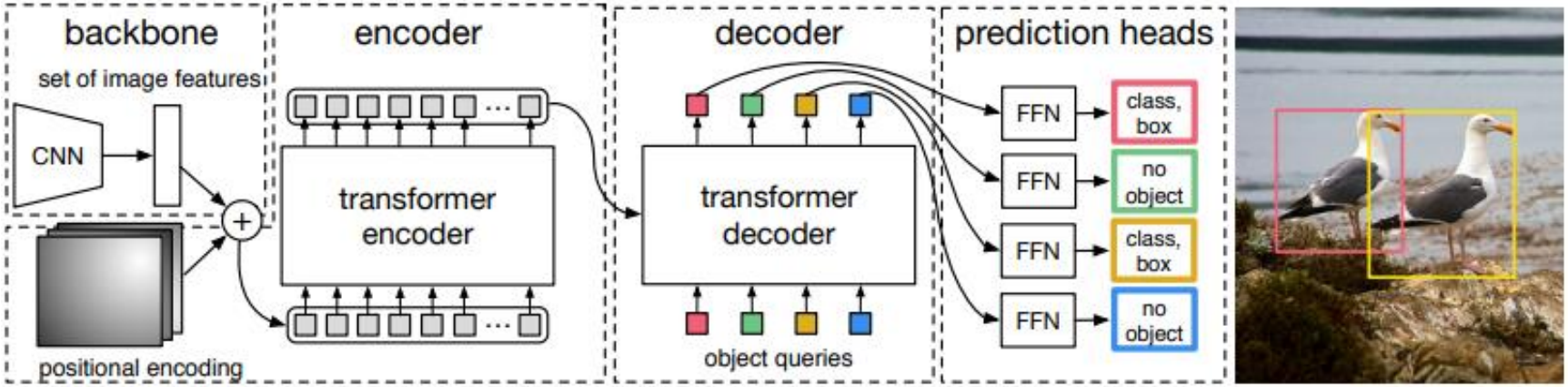
Self-Attention GAN



https://arxiv.org/abs/1805.08318

這裡給兩個使用self-attention在影像上的例子

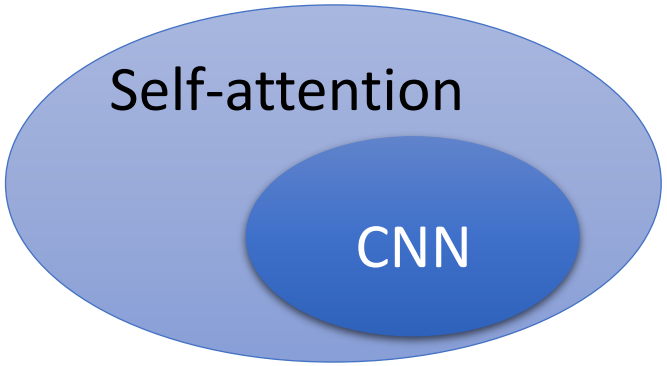
DEtection Transformer (DETR)

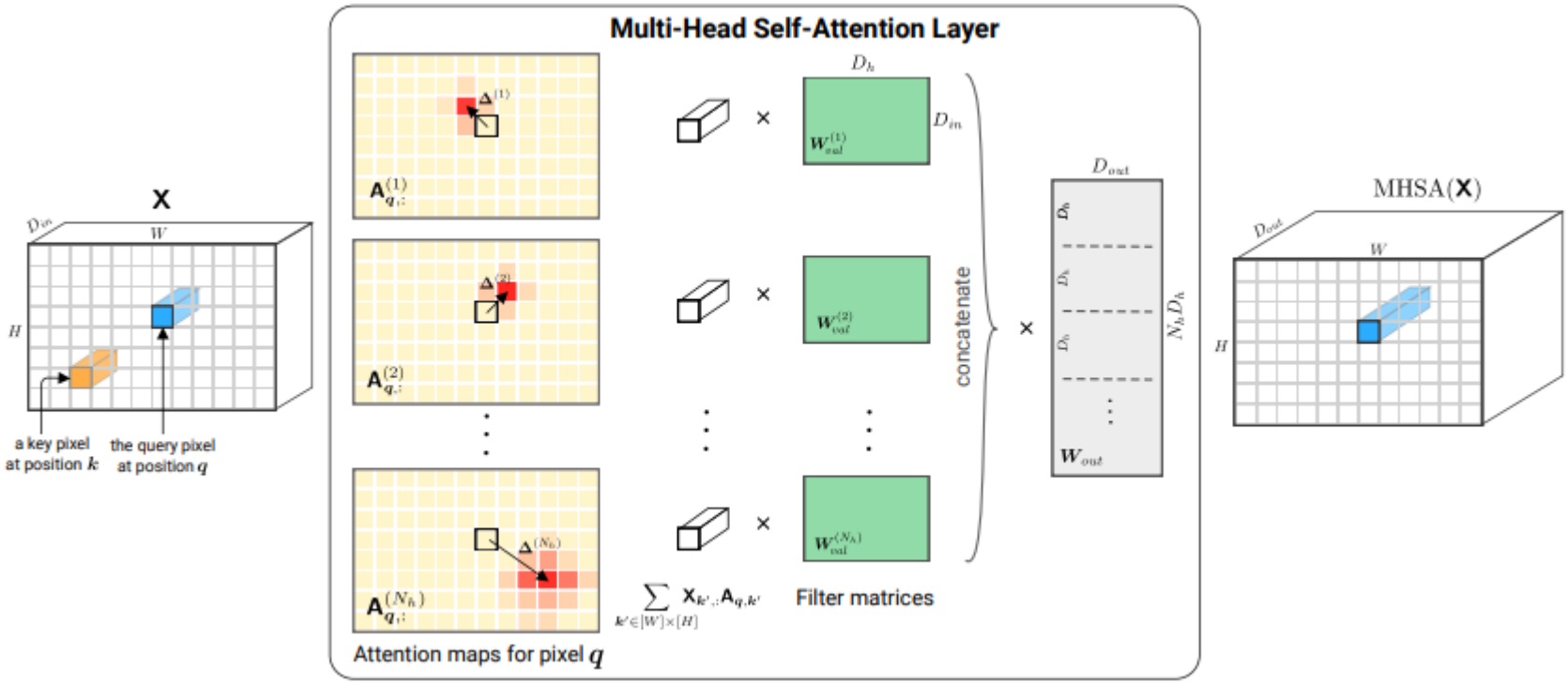


https://arxiv.org/abs/2005.12872

看v8的投影片，這份的有少

Self-attention v.s. CNN





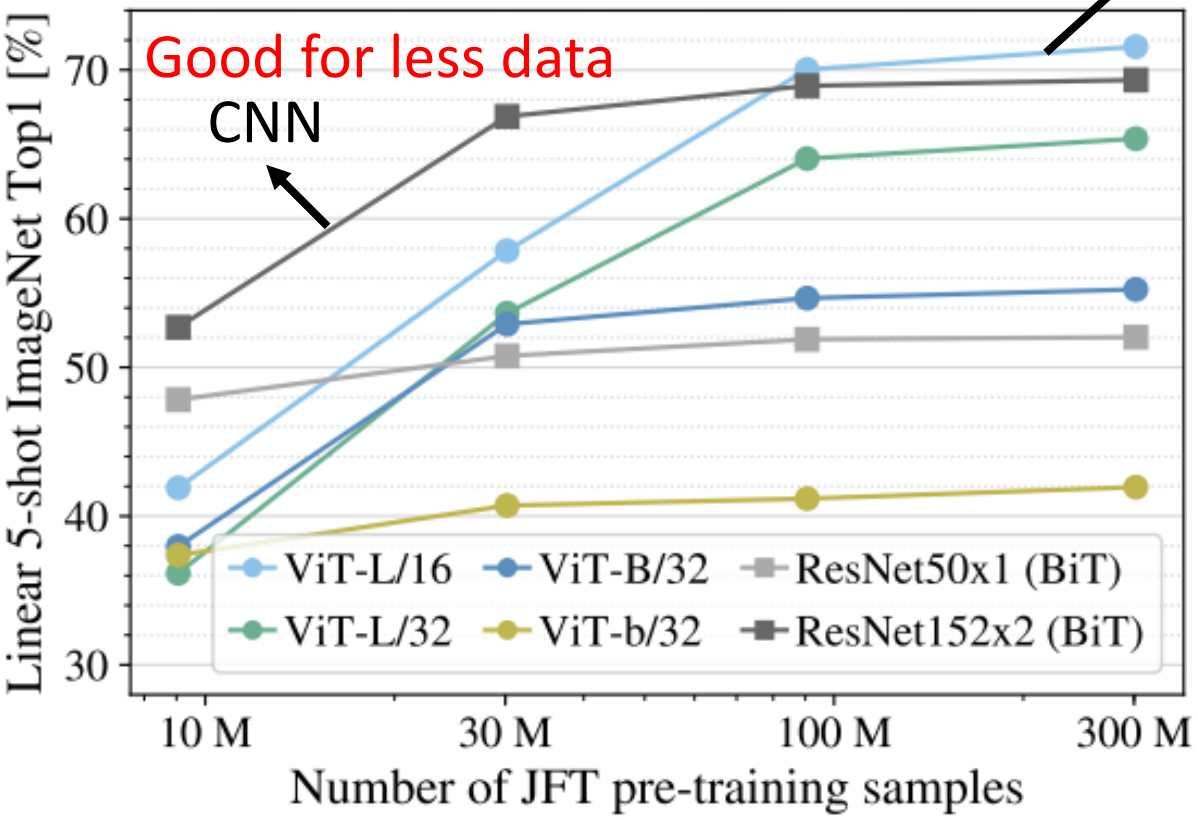
On the Relationship between Self-Attention and Convolutional Layers

https://arxiv.org/abs/1911.03584

Self-attention v.s. CNN

Good for more data

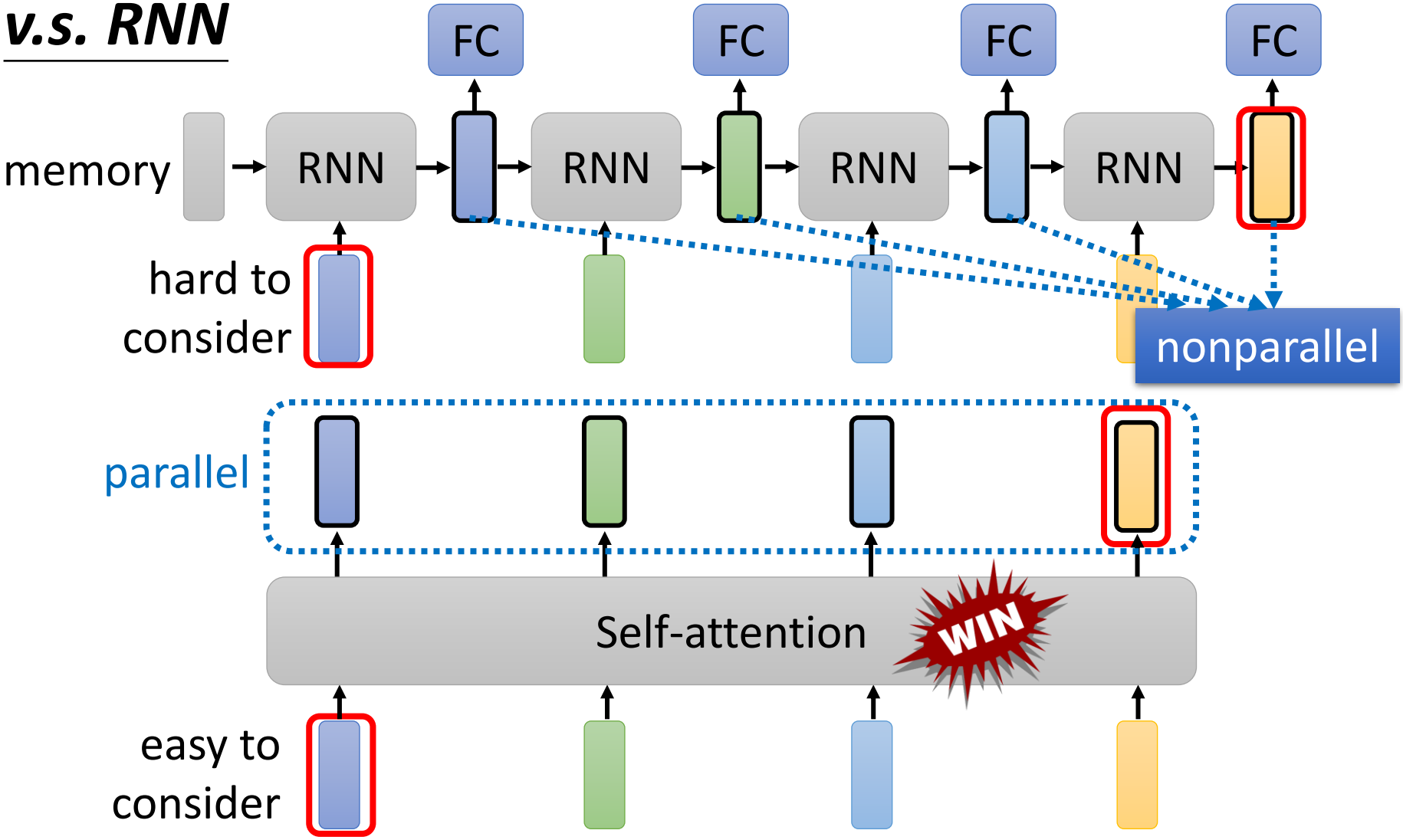
Self-attention



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

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

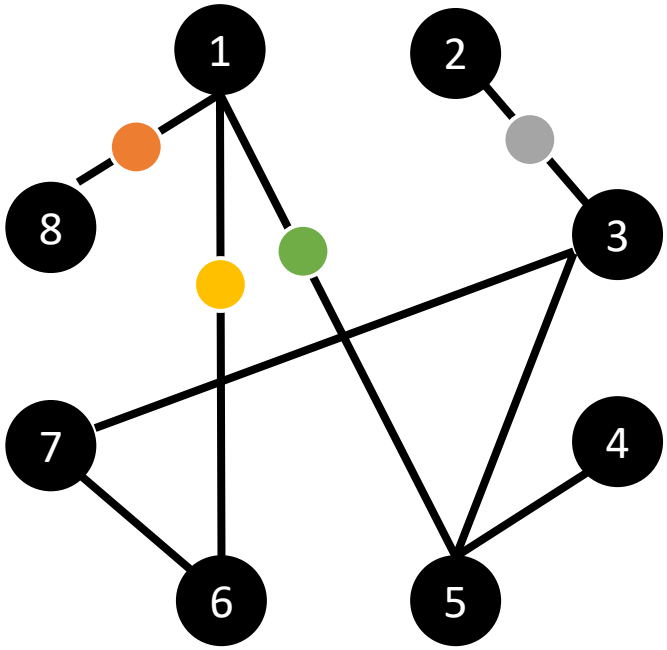
Self-attention v.s. RNN



Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention

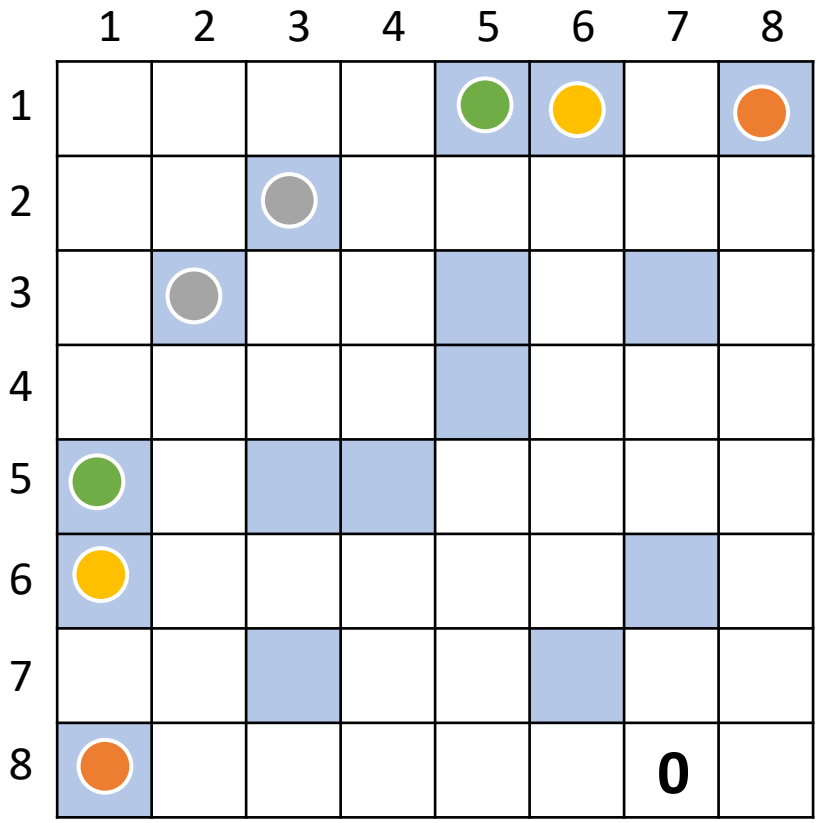
https://arxiv.org/abs/2006.16236

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)

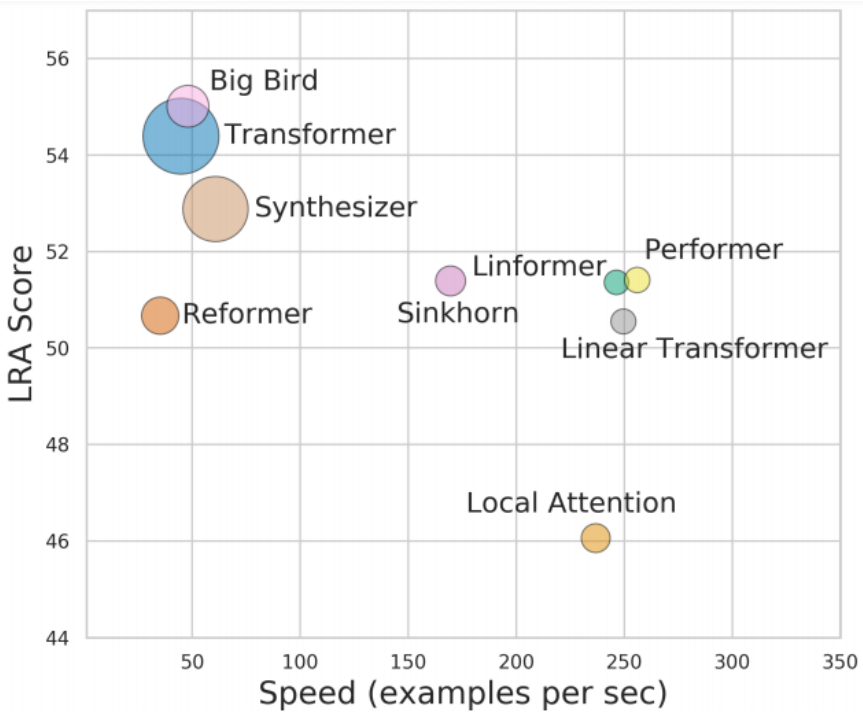
https://youtu.be/M9ht8vsVEw8

(in Mandarin)

To Learn More …

Long Range Arena: A Benchmark for Efficient Transformers

https://arxiv.org/abs/2011.04006



Efficient Transformers: A Survey

https://arxiv.org/abs/2009.06732