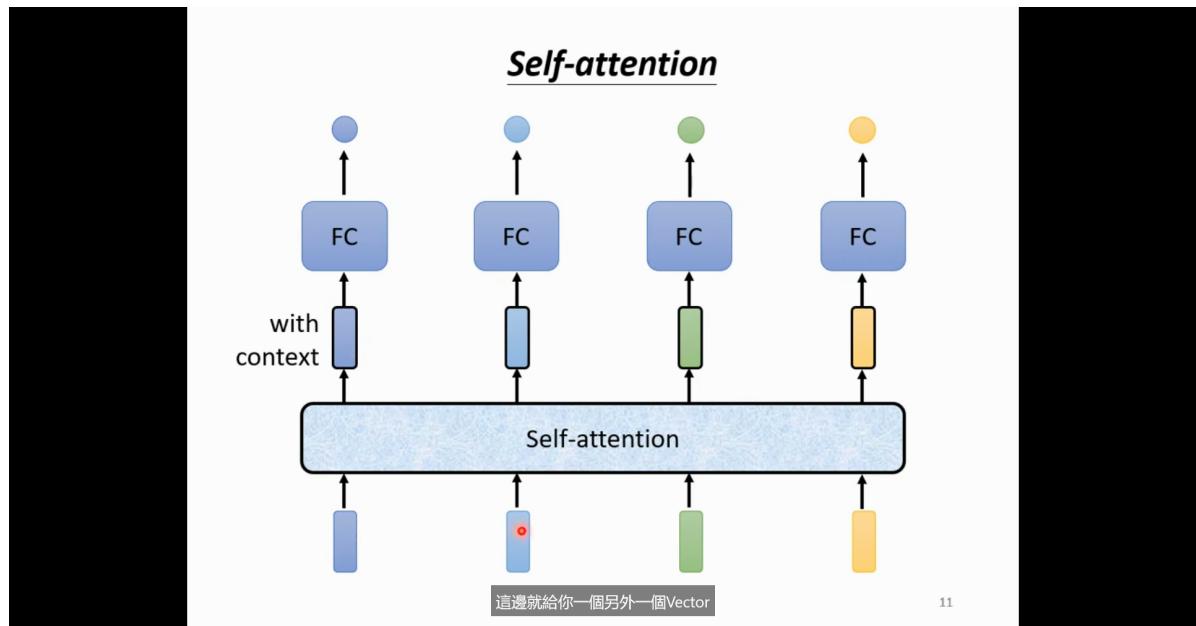


输入几个向量输出几个label

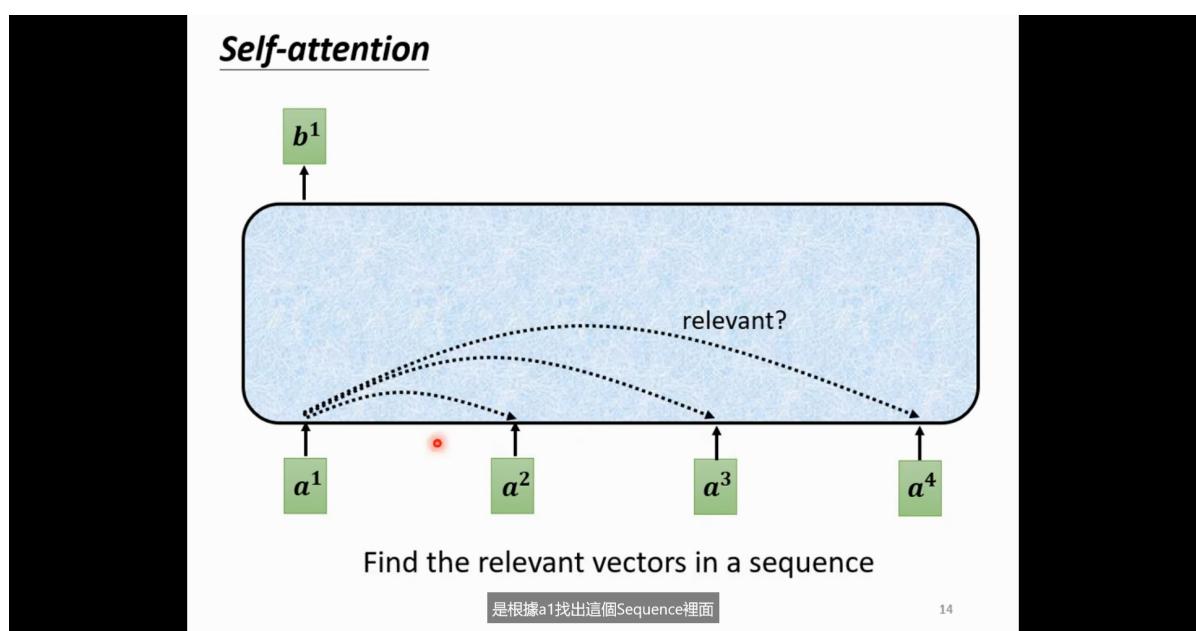


输出的4个都考虑了整个输入序列

可以叠好多层

怎样计算b1呢?

## Self-Attention



1. 根据 $a^1$ 找到输入序列中跟 $a^1$ 相关的向量, 每个其他的输入跟 $a^1$ 相关的程度用 $\alpha$ 来表示

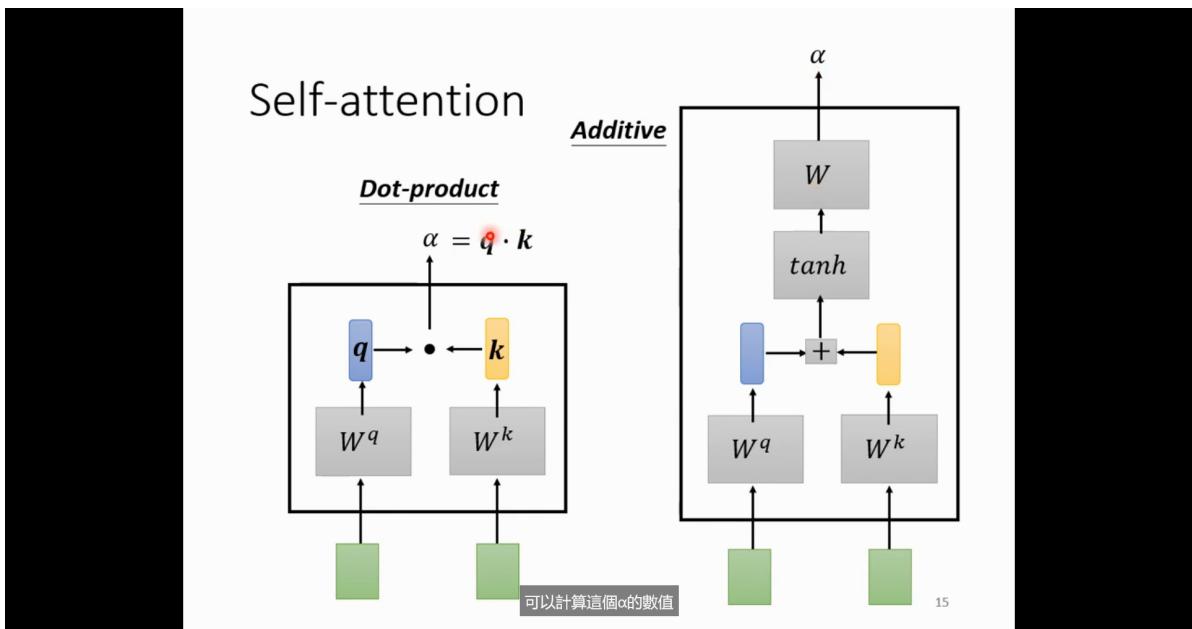
下图为计算权重系数的方法

两个输入向量 分别乘权重矩阵, 得到 $q, k$

$$q = a * W^q$$

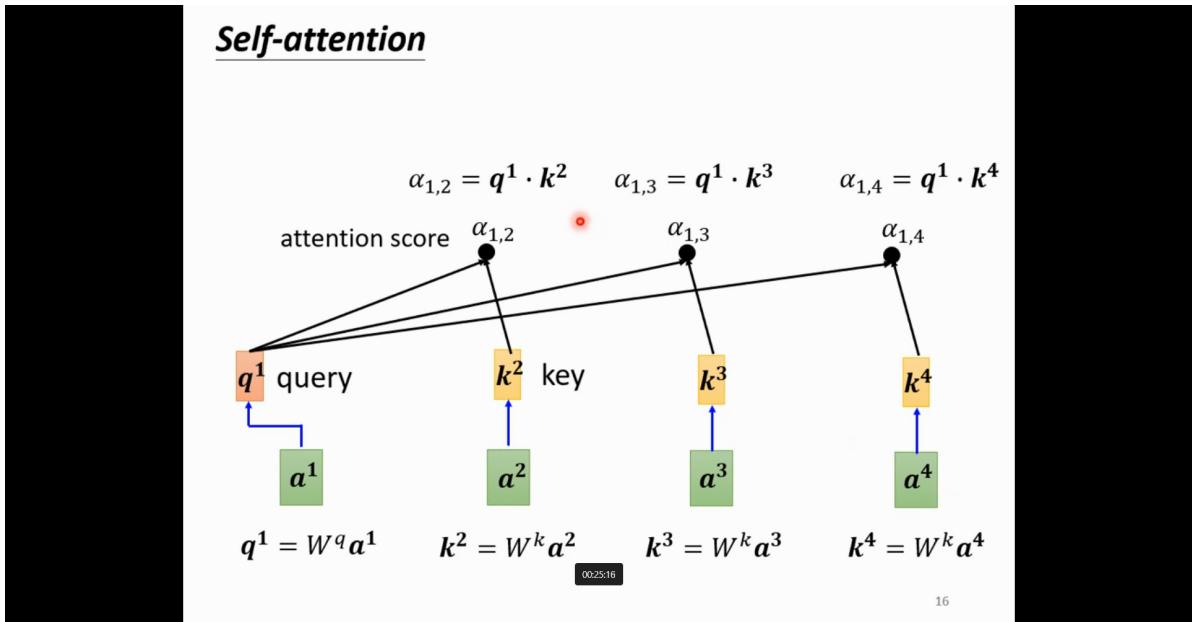
$$k = a * W^k$$

$q, k$ 再内积则为权重系数

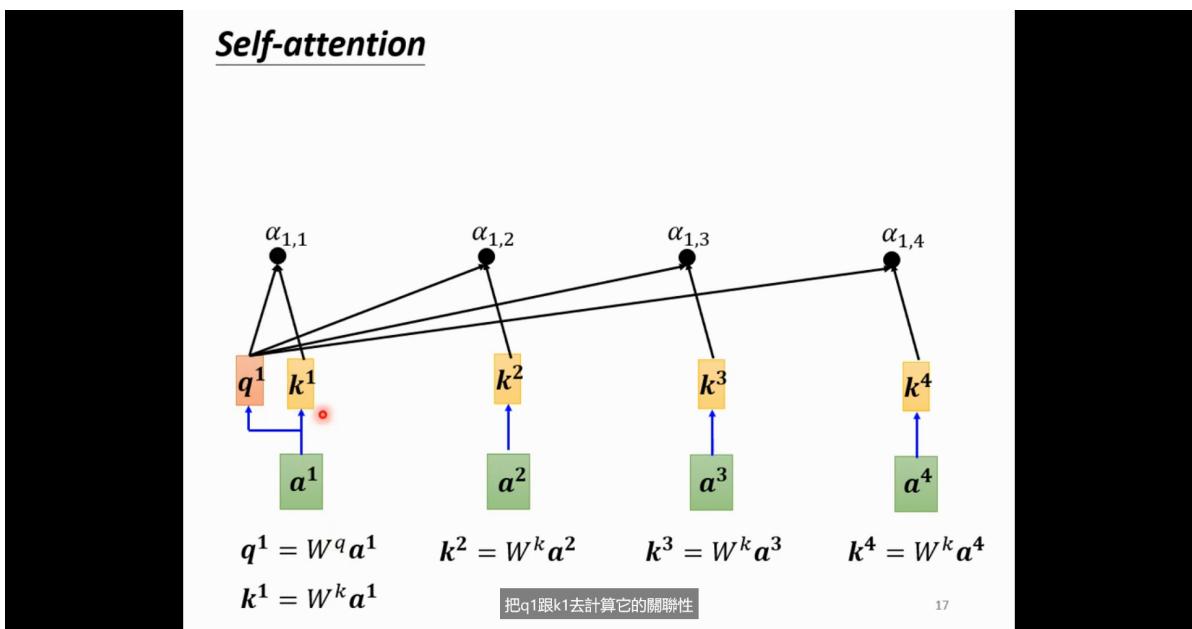


$\alpha_{1,2}$  表示  $\mathbf{q}$  是  $\mathbf{a}_1$  提供的  $\mathbf{k}$  是  $\mathbf{a}_2$  提供的

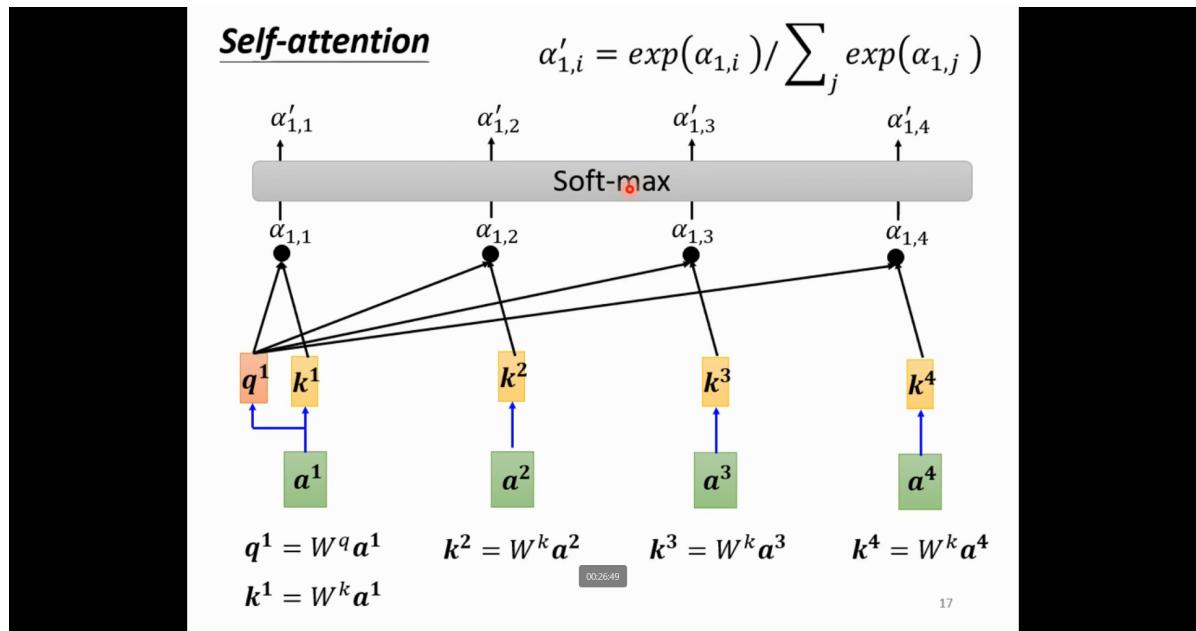
用  $\mathbf{a}_1$  分别去和  $\mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$  内积，算出 3 个注意力系数



一般情况， $\mathbf{a}_1$  也要和自己算一个相关性



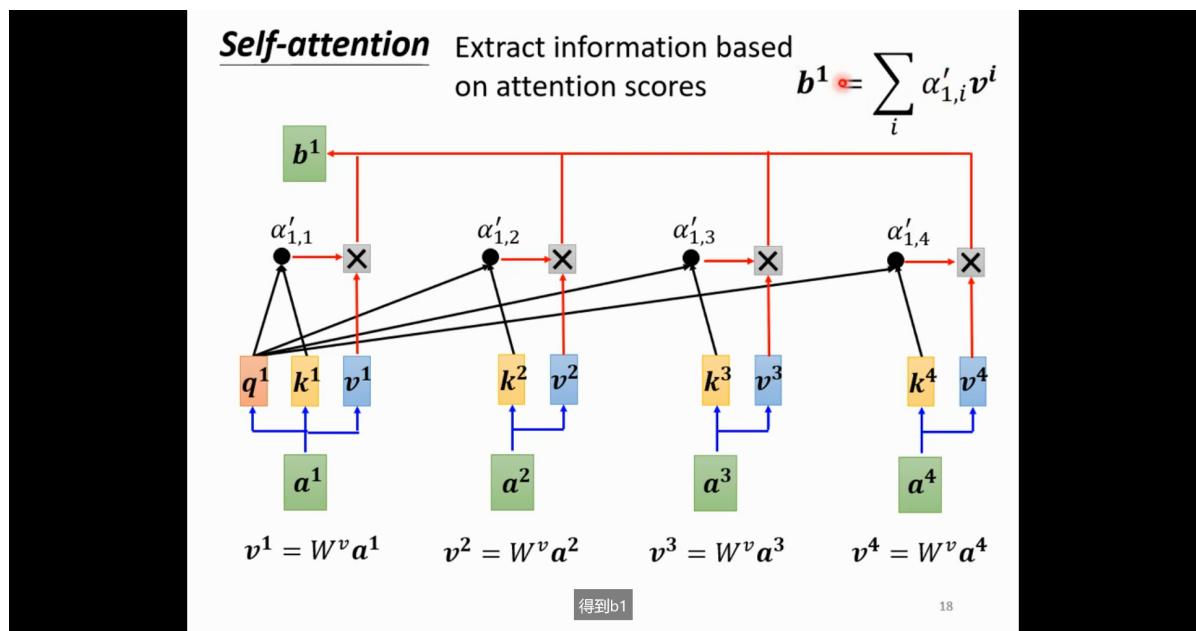
经过softmax得到 $\alpha_{1,2}^{'}$



得到 $\alpha_{1,2}^{'}$ 后会根据权重抽取信息

把输入乘上 $W^v$ 后得到 $v$

$v$ 再乘上 $\alpha_{1,2}^{'}$ 后,所有的相加即为 $b_1$



$b_1$ 到 $b_4$ 是同时被计算出来的

## 矩阵乘法

可以把 $a_1, a_2, a_3, a_4$ 拼起来(分别为列)变成一个矩阵 $W$ 相乘

$$q^i = W^q a^i \quad q^1 q^2 q^3 q^4 = \begin{matrix} Q \\ W^q \end{matrix} \quad a^1 a^2 a^3 a^4 = \begin{matrix} I \\ W^q \end{matrix}$$

$$k^i = W^k a^i \quad k^1 k^2 k^3 k^4 = \begin{matrix} K \\ W^k \end{matrix} \quad a^1 a^2 a^3 a^4 = \begin{matrix} I \\ W^k \end{matrix}$$

$$v^i = W^v a^i \quad v^1 v^2 v^3 v^4 = \begin{matrix} V \\ W^v \end{matrix} \quad a^1 a^2 a^3 a^4 = \begin{matrix} I \\ W^v \end{matrix}$$

得到的  $k$  都变成行向量和  $q$  相乘分别得到  $\alpha$

$\alpha'_{1,1}$	$\alpha'_{2,1}$	$\alpha'_{3,1}$	$\alpha'_{4,1}$	$\alpha'_{1,1}$	$\alpha'_{2,1}$	$\alpha'_{3,1}$	$\alpha'_{4,1}$	$k^1$
$\alpha'_{1,2}$	$\alpha'_{2,2}$	$\alpha'_{3,2}$	$\alpha'_{4,2}$	$\alpha'_{1,2}$	$\alpha'_{2,2}$	$\alpha'_{3,2}$	$\alpha'_{4,2}$	$k^2$
$\alpha'_{1,3}$	$\alpha'_{2,3}$	$\alpha'_{3,3}$	$\alpha'_{4,3}$	$\alpha'_{1,3}$	$\alpha'_{2,3}$	$\alpha'_{3,3}$	$\alpha'_{4,3}$	$k^3$
$\alpha'_{1,4}$	$\alpha'_{2,4}$	$\alpha'_{3,4}$	$\alpha'_{4,4}$	$\alpha'_{1,4}$	$\alpha'_{2,4}$	$\alpha'_{3,4}$	$\alpha'_{4,4}$	$k^4$
$A'$	softmax 你會對這邊的每一個 column				$Q$	$K^T$	23	

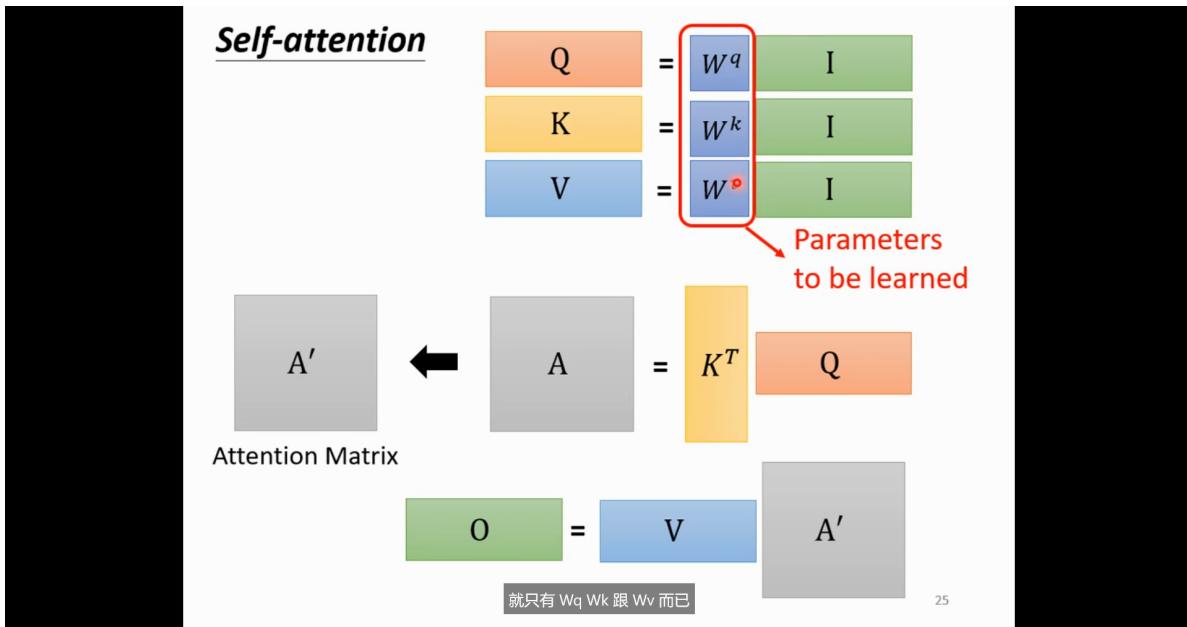
每列做 SoftMax

再乘上  $v$

$b^1$	$=$	$v^1 v^2 v^3 v^4$	$A'$
			$\alpha'_{1,1} \alpha'_{2,1} \alpha'_{3,1} \alpha'_{4,1}$
			$\alpha'_{1,2} \alpha'_{2,2} \alpha'_{3,2} \alpha'_{4,2}$
			$\alpha'_{1,3} \alpha'_{2,3} \alpha'_{3,3} \alpha'_{4,3}$
			$\alpha'_{1,4} \alpha'_{2,4} \alpha'_{3,4} \alpha'_{4,4}$
全部加起來得到 $b_1$			

$b^1 b^2 b^3 b^4$	$=$	$v^1 v^2 v^3 v^4$	$A'$
0			$\alpha'_{1,1} \alpha'_{2,1} \alpha'_{3,1} \alpha'_{4,1}$
			$\alpha'_{1,2} \alpha'_{2,2} \alpha'_{3,2} \alpha'_{4,2}$
			$\alpha'_{1,3} \alpha'_{2,3} \alpha'_{3,3} \alpha'_{4,3}$
			$\alpha'_{1,4} \alpha'_{2,4} \alpha'_{3,4} \alpha'_{4,4}$
就是 Self-attention 的輸出			

综合一下，只要学习  $W$

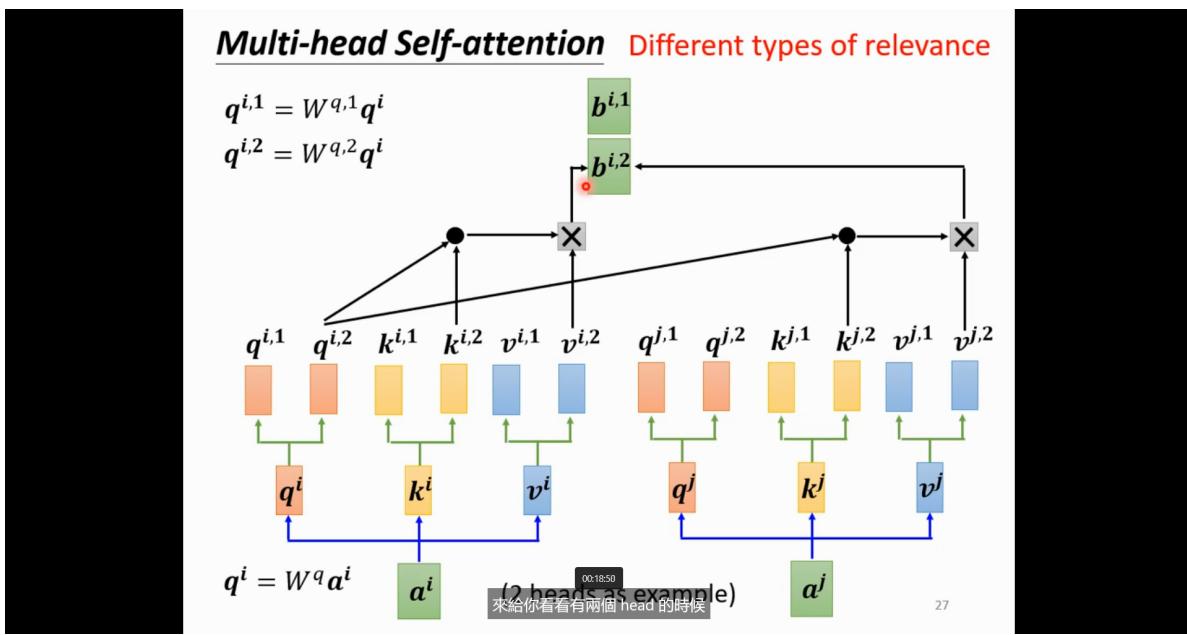


## Multi-head Self-attention

1的那一类自己做attention

2的那一类自己做attention

下图为2head,就是一个q,k,v分别乘两个举证变成了两类q,k,v



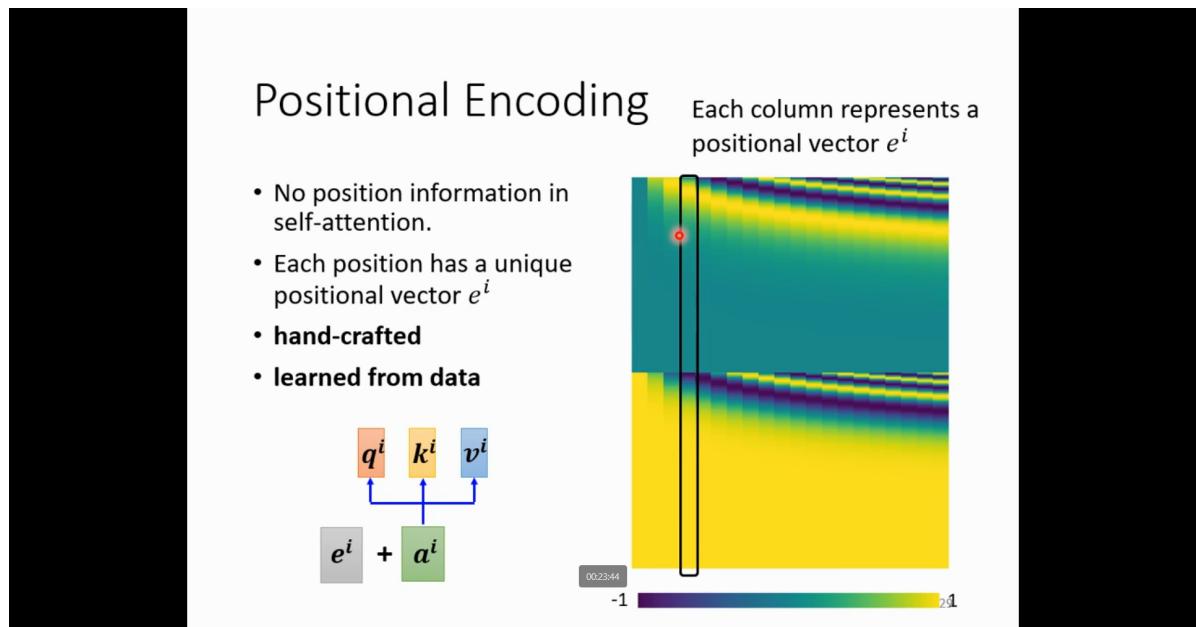
$$b^i = W^o$$

$b^{i,1}$   
 $b^{i,2}$

# Position Encoder

Self-attention是缺少位置信息,每个都乘了其余的信息

为每个输入都设置一个positional vector  $e^i$ 来代表位置信息,并加到 $a^i$ 上

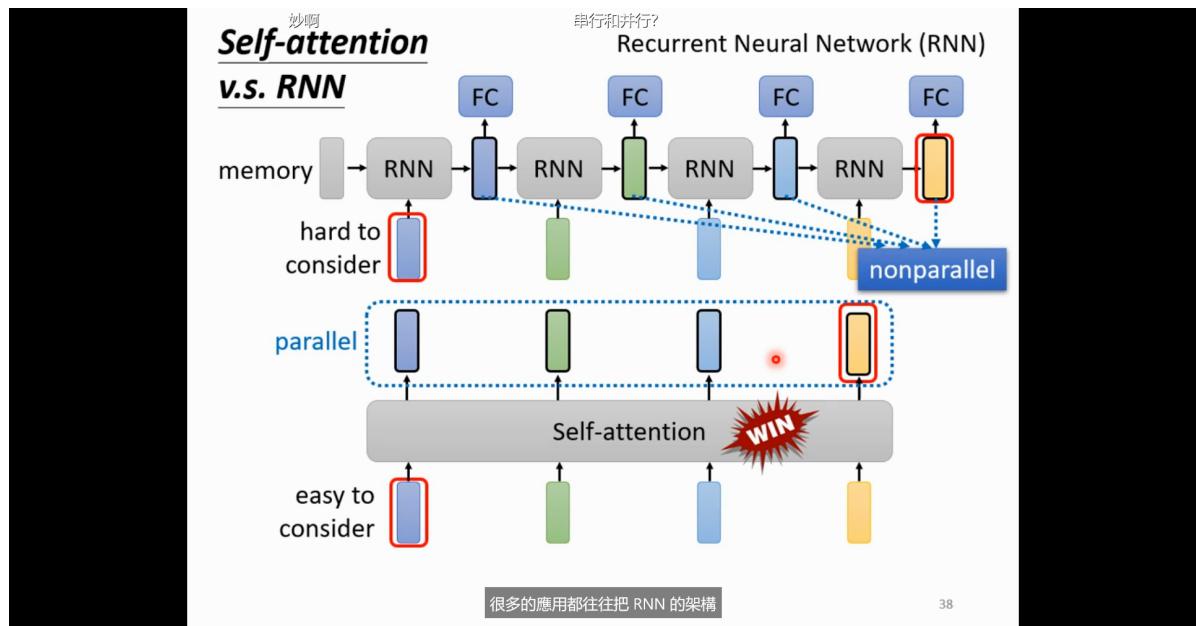


右边每一列表示每个输入的位置信息

## Tips

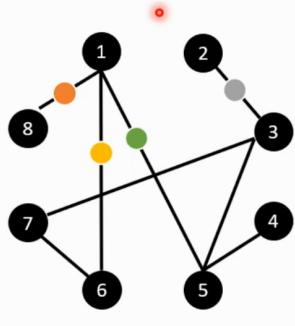
Self-Attention也可以不看整个句子,可能一个小的范围就可以

## RNN和Self-Attention区别



## Self-Attention与Graph

## Self-attention for Graph



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

		Attention Matrix							
		1	2	3	4	5	6	7	8
1	1								
	2								
3									
4									
5									
6									
7									
8									0

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

所有 GNN 的各種變形了

40