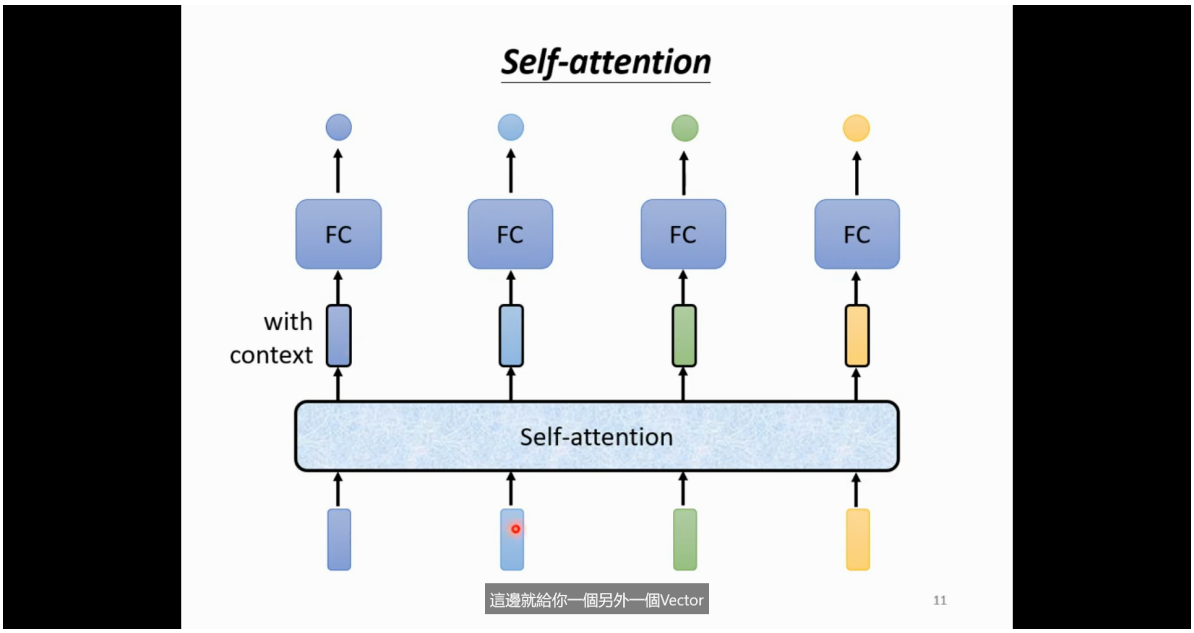
输入几个向量输出几个lable

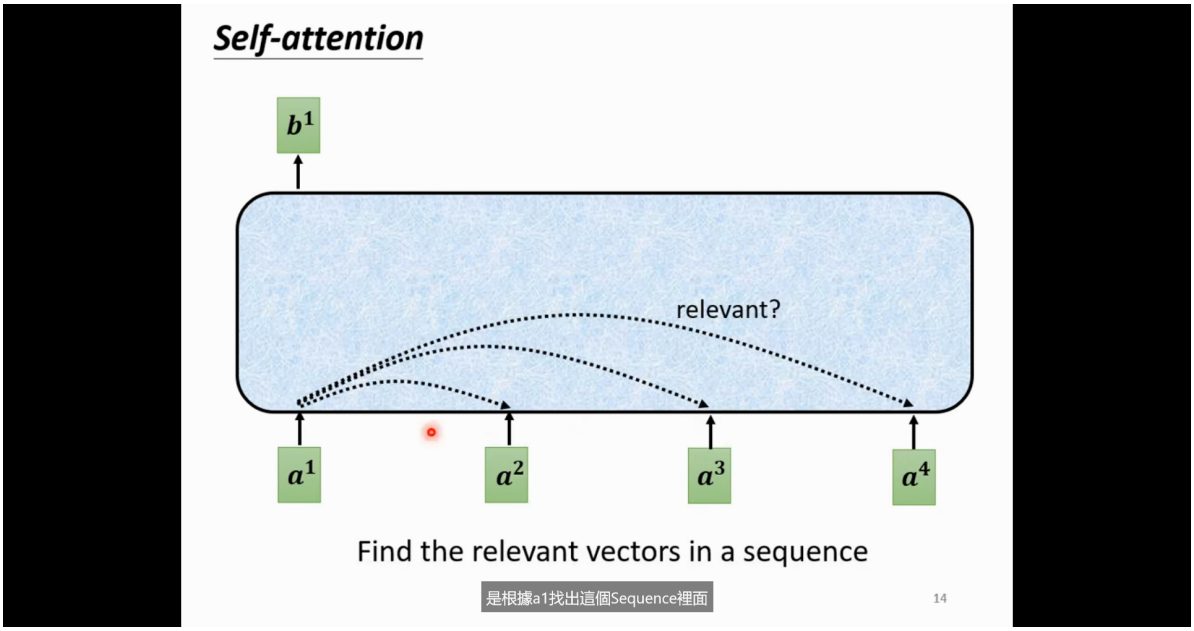


输出的4个都考虑了整个输入序列

可以叠好多层

怎样计算b1呢?

Self-Attention



1. 根据a1找到输入序列中跟a1相关的向量,每个其他的输入跟a1相关的程度用α来表示

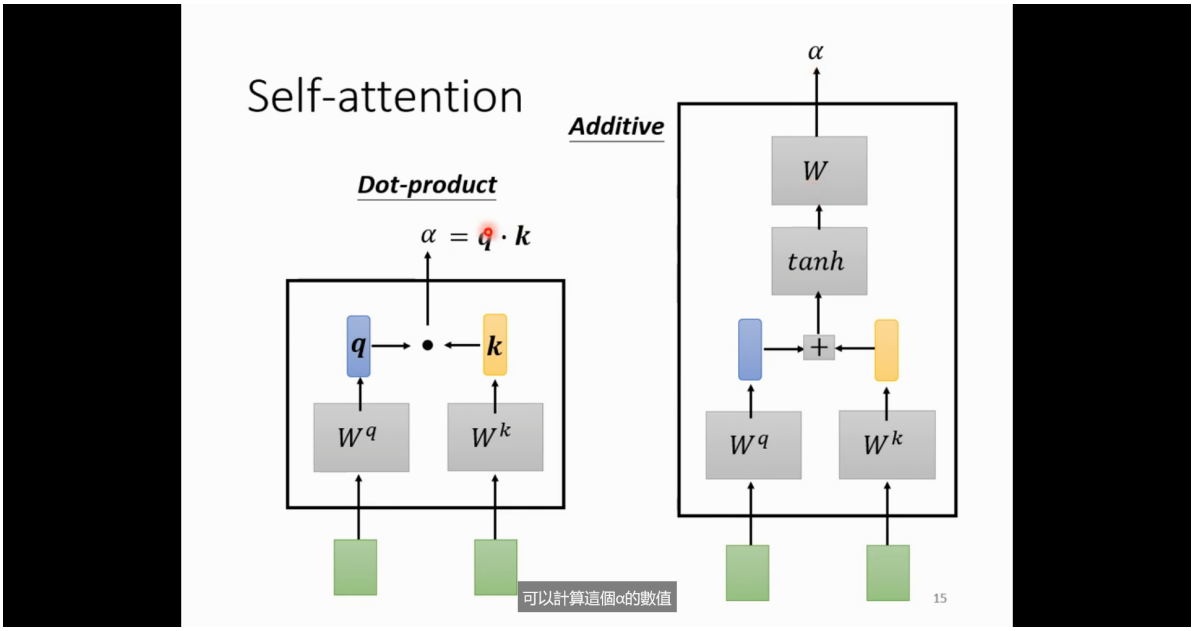
   下图为计算权重系数的方法

   两个输入向量 分别乘权重矩阵,得到q,k

   q = $a * W^{q}$

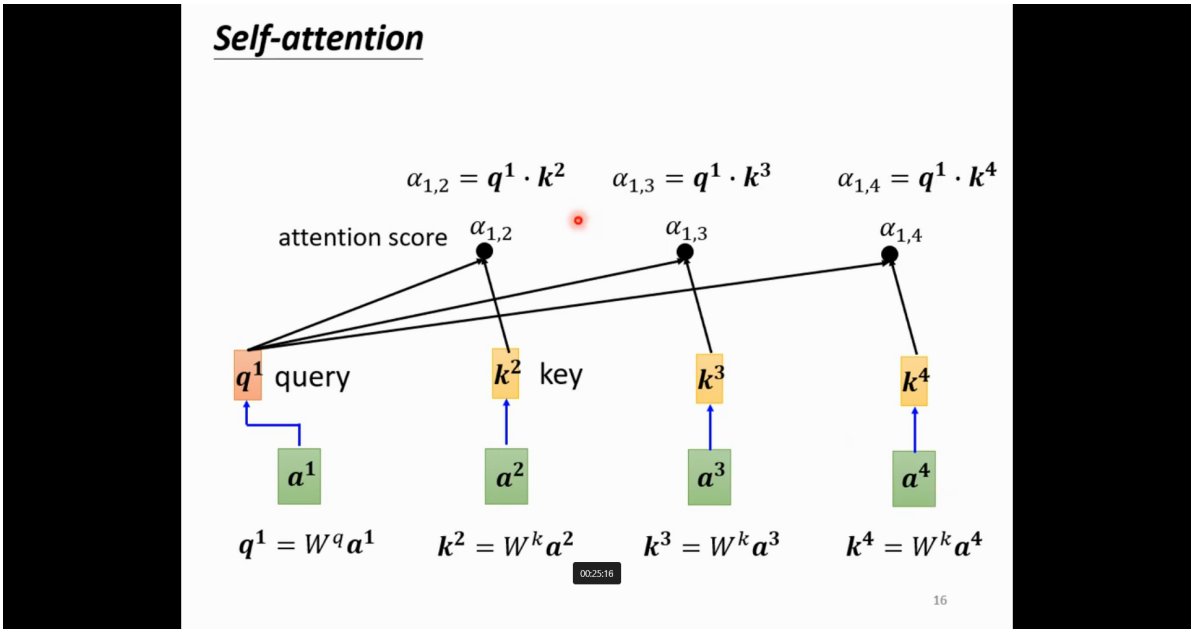
   v = $a * W^{v}$

   q,k再内积则为权重系数

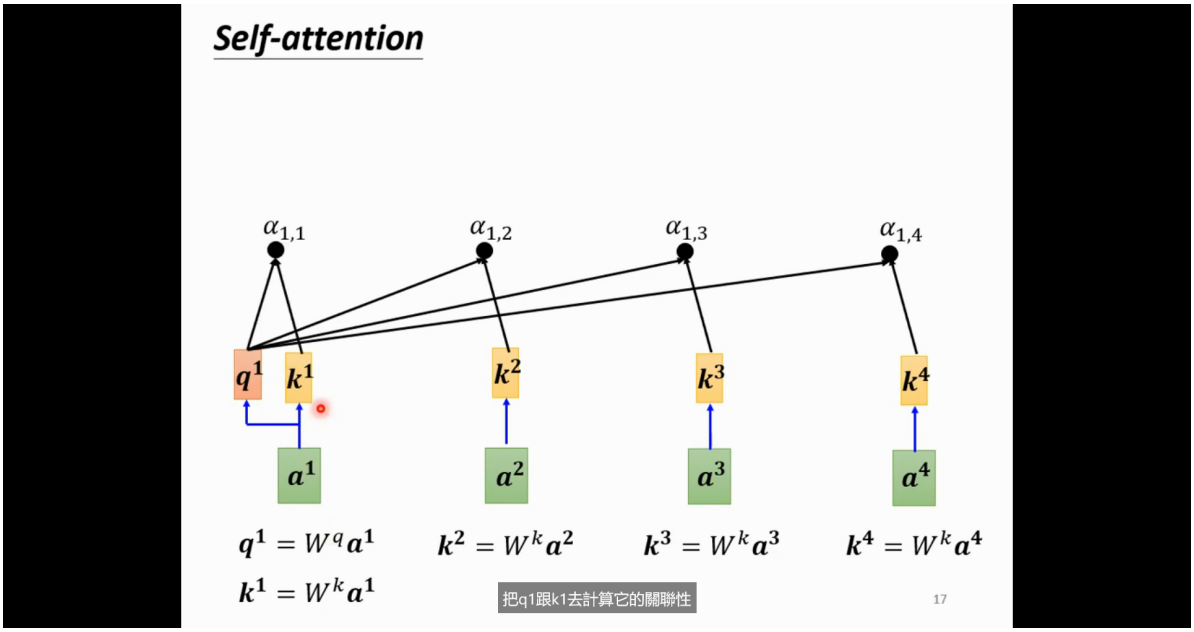


$\alpha_{1,2}$表示q是1提供的k是2提供的

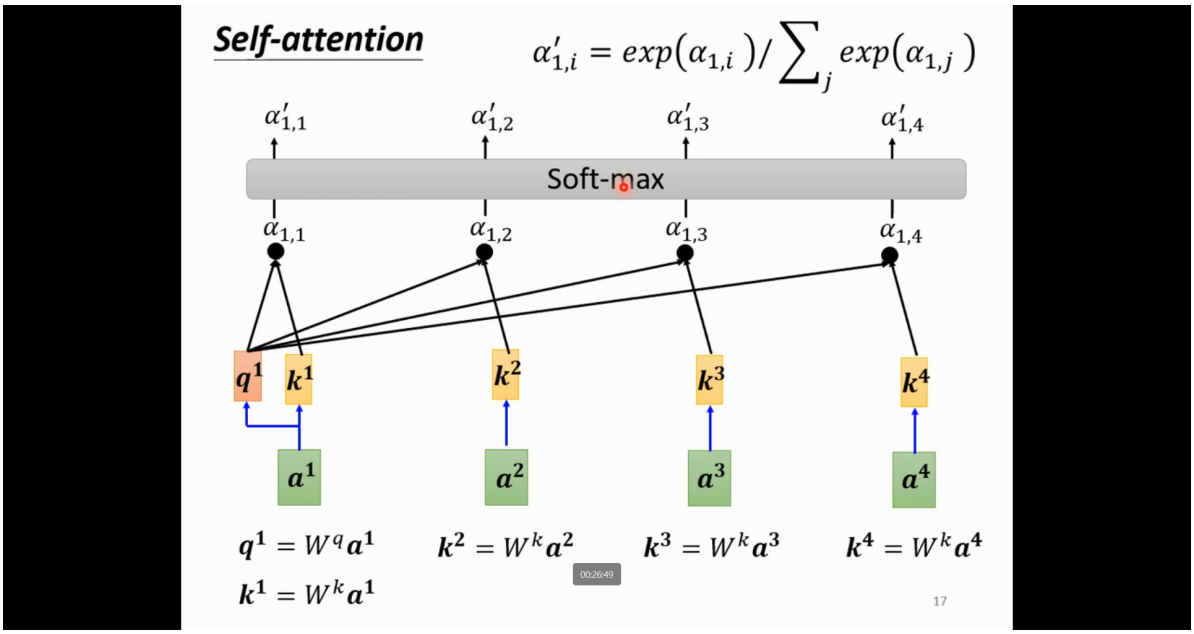
用a1分别去和a2,a3,a4内积,算出3个注意力系数



一般情况,a1也要和自己算一个相关性



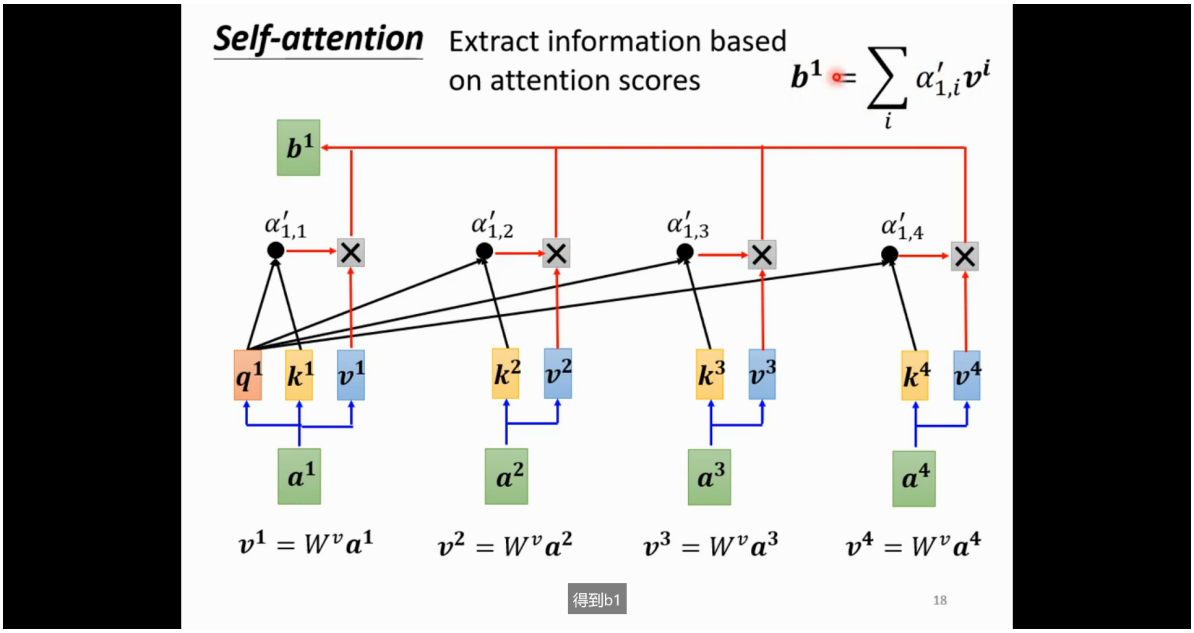
经过softmax得到$\alpha_{1,2}^{'}$



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

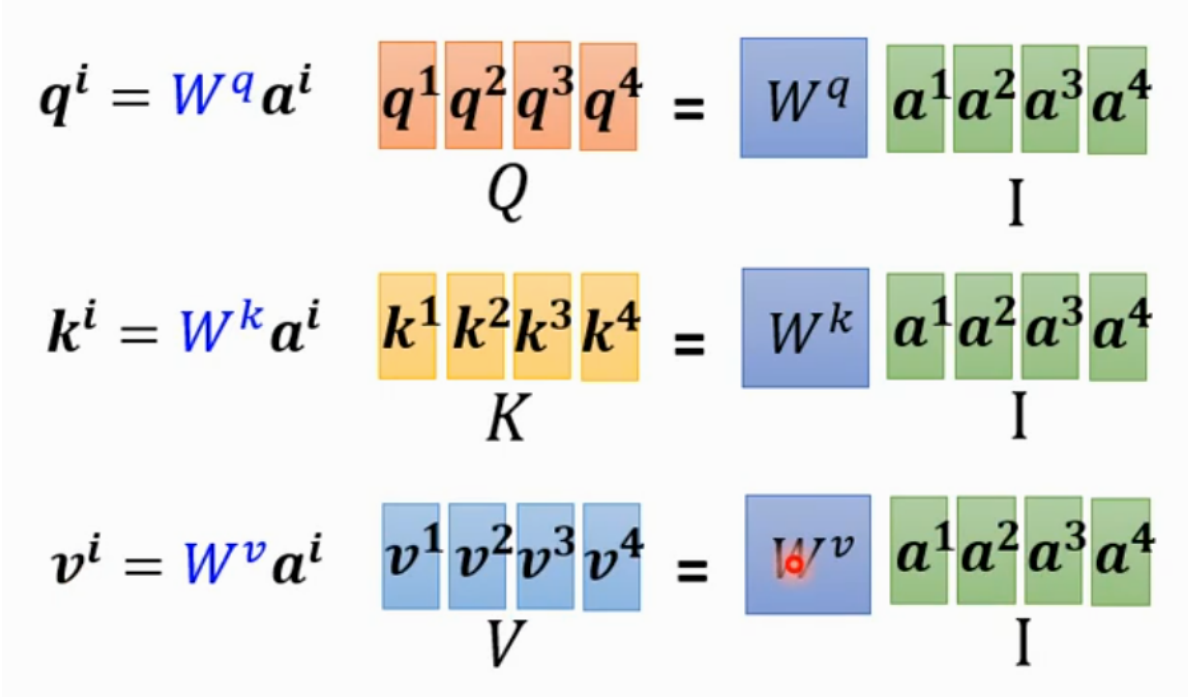
v再乘上$\alpha_{}^{'}$后,所有的相加即为b1



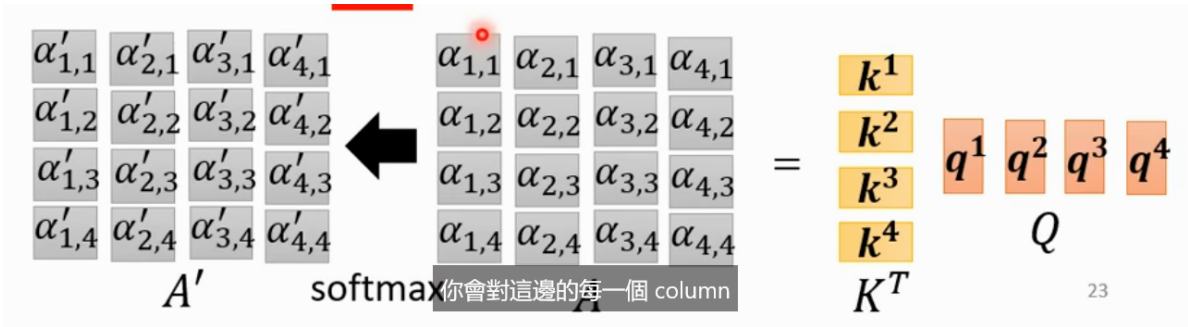
b1到b4是同时被计算出来的

矩阵乘法

可以把a1,a2,a3,a4拼起来(分别为列)变成一个矩阵W相乘

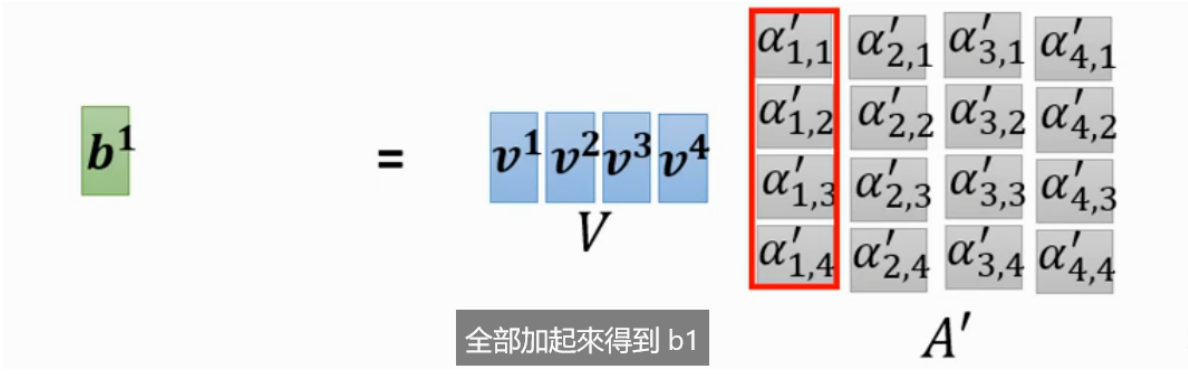


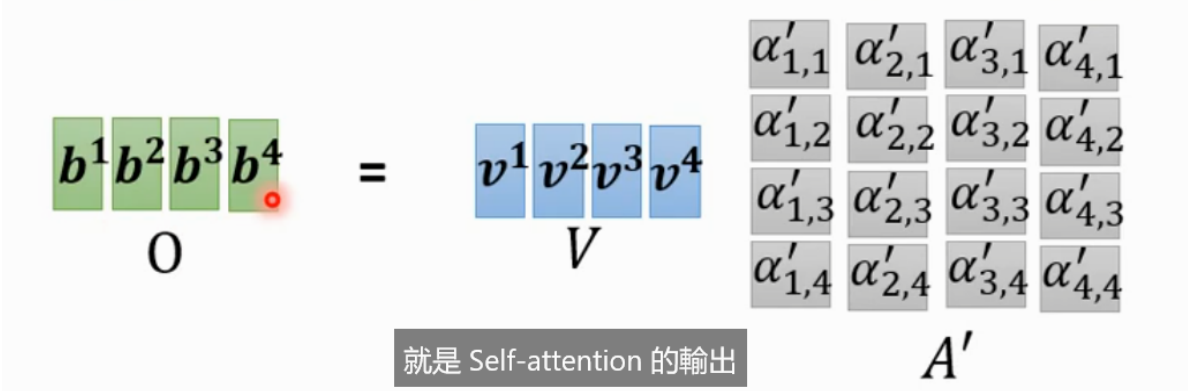
得到的k都变成行向量和q相乘分别得到α



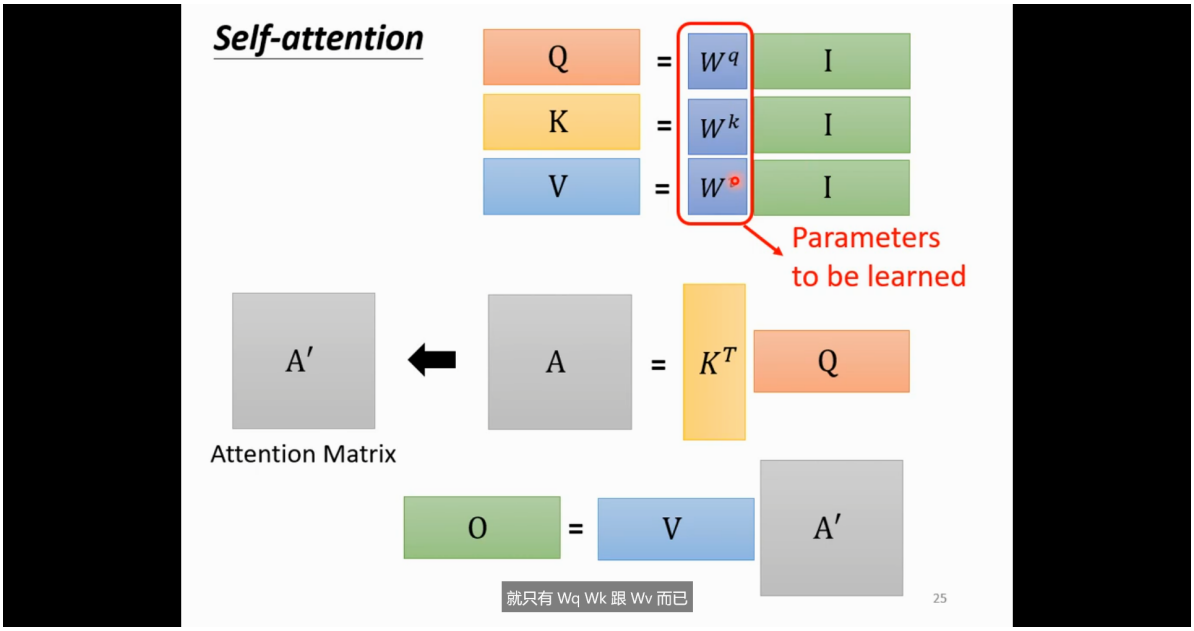
每列做SoftMax

再乘上v





综合一下,只要学习W

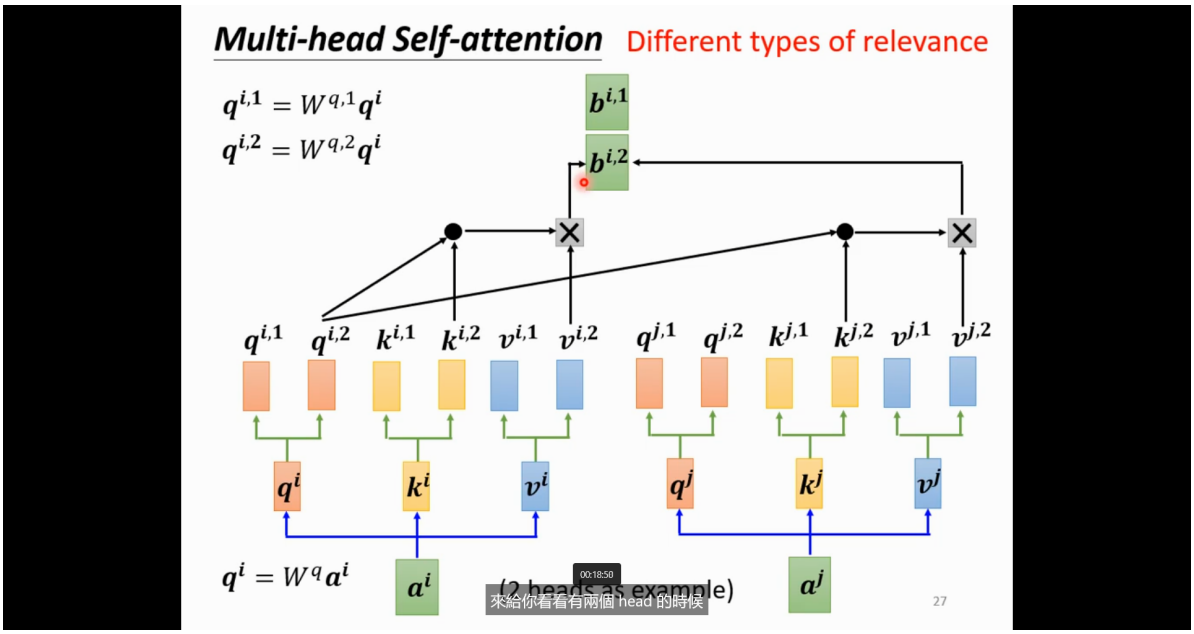


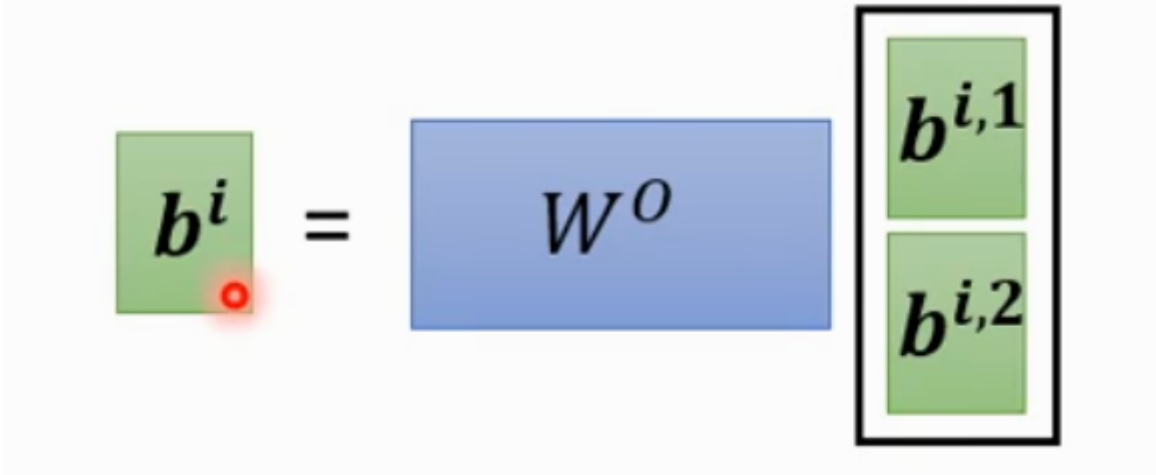
Multi-head Self-attention

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

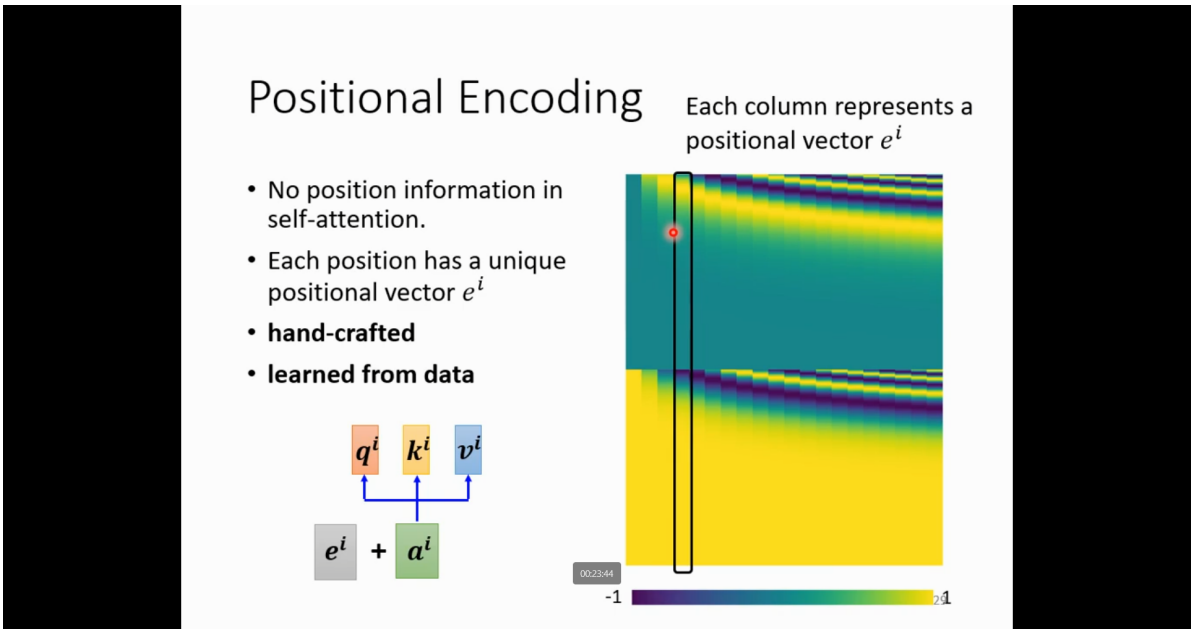




Position Enconder

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

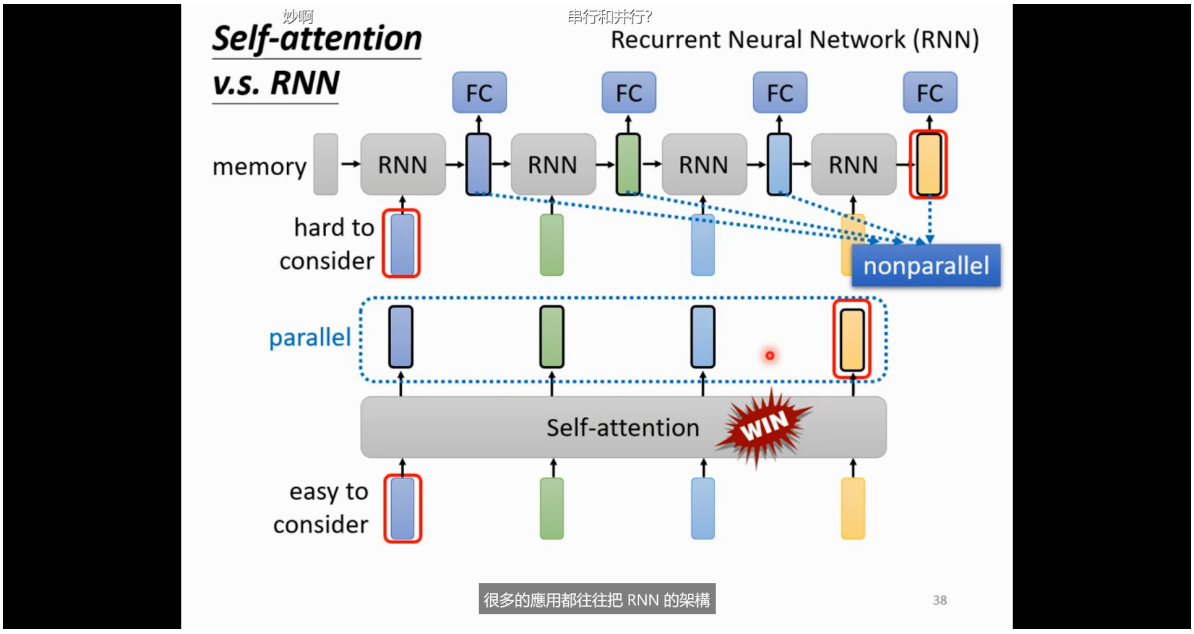


右边每一列表示每个输入的位置信息

Tips

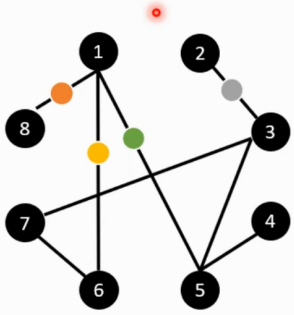
Self-Attention也可以不看整个句子,可能一个小的范围就可以

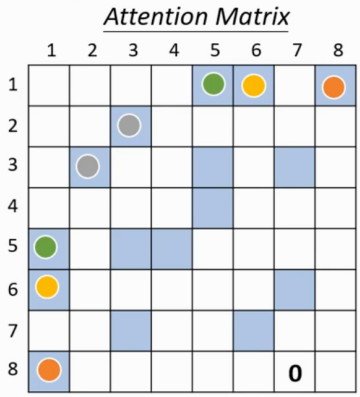
RNN和Self-Attention区别



Self-Attention与Graph

Self-attention for Graph





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

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

所有 GNN 的各種變形了

老师的门铃响了