

Self-attention

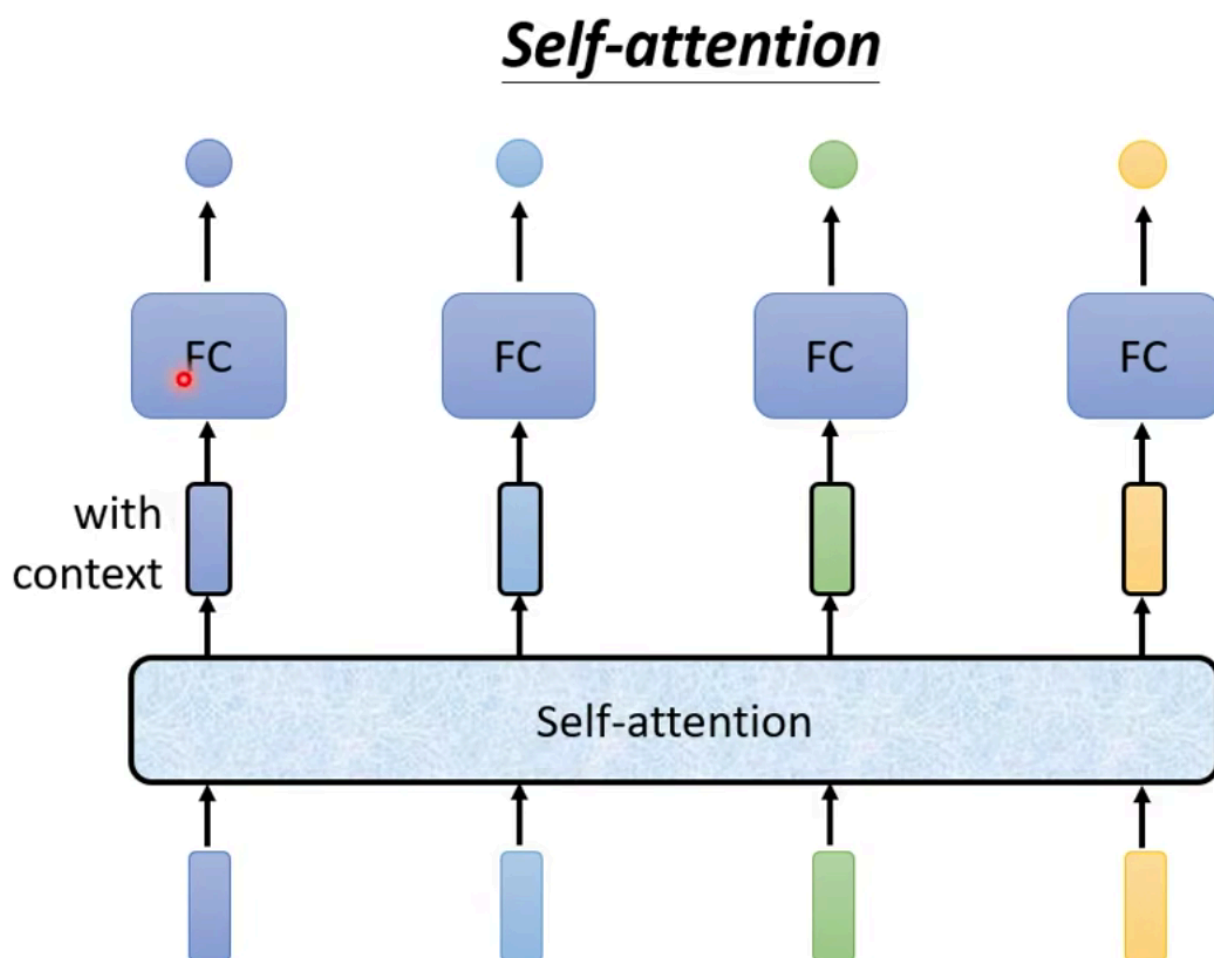
ATTENTION

1. 输入与输出

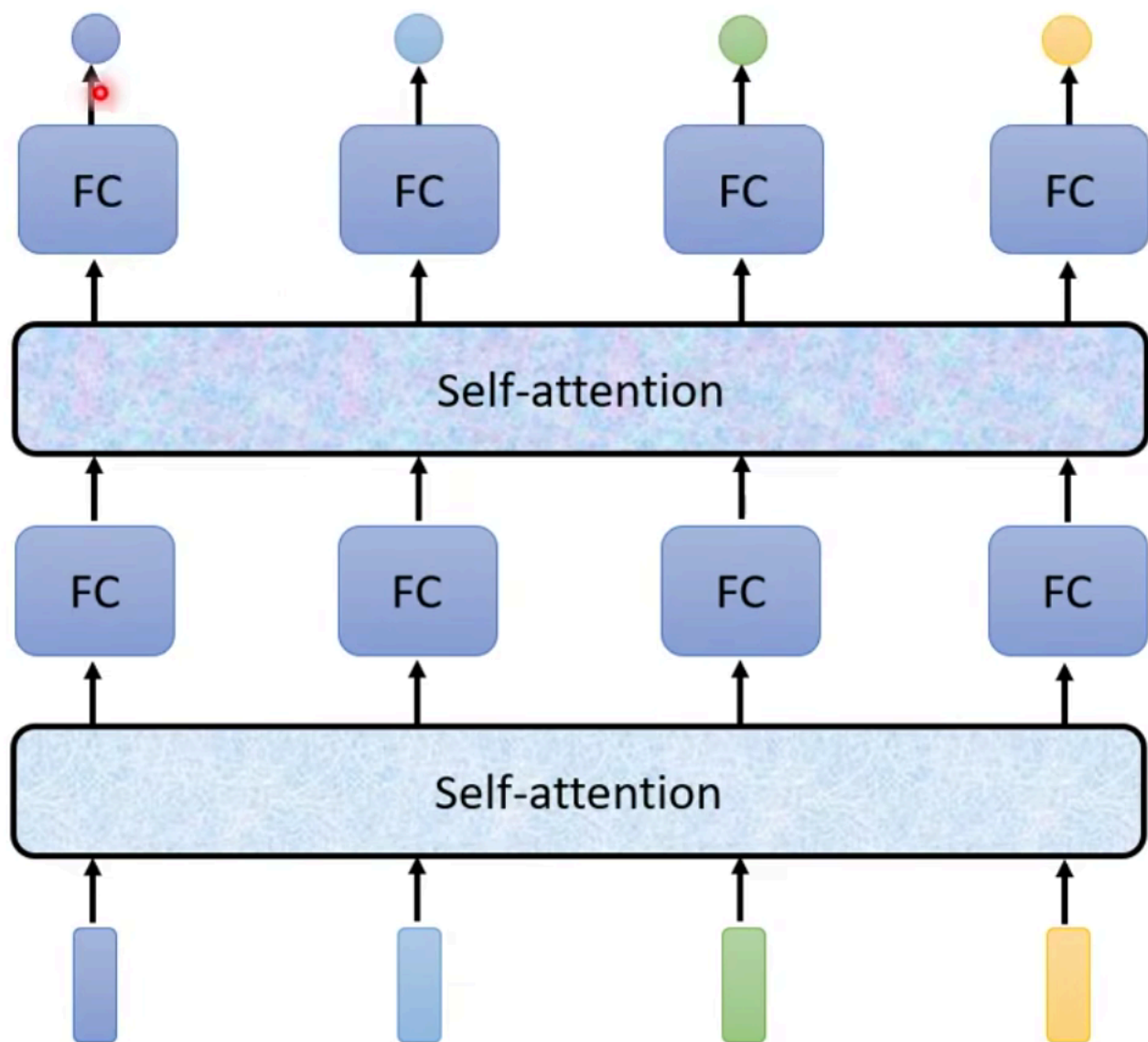
- 输入：一堆向量
- 输出：
 1. 每一个向量都有一个label (输入输出数目一样->Sequence labeling)
 2. 整个序列只有一个label
 3. 模型自己决定了输出的label的数量 (seq2seq)

2. Sequence Labeling

2.1 机制



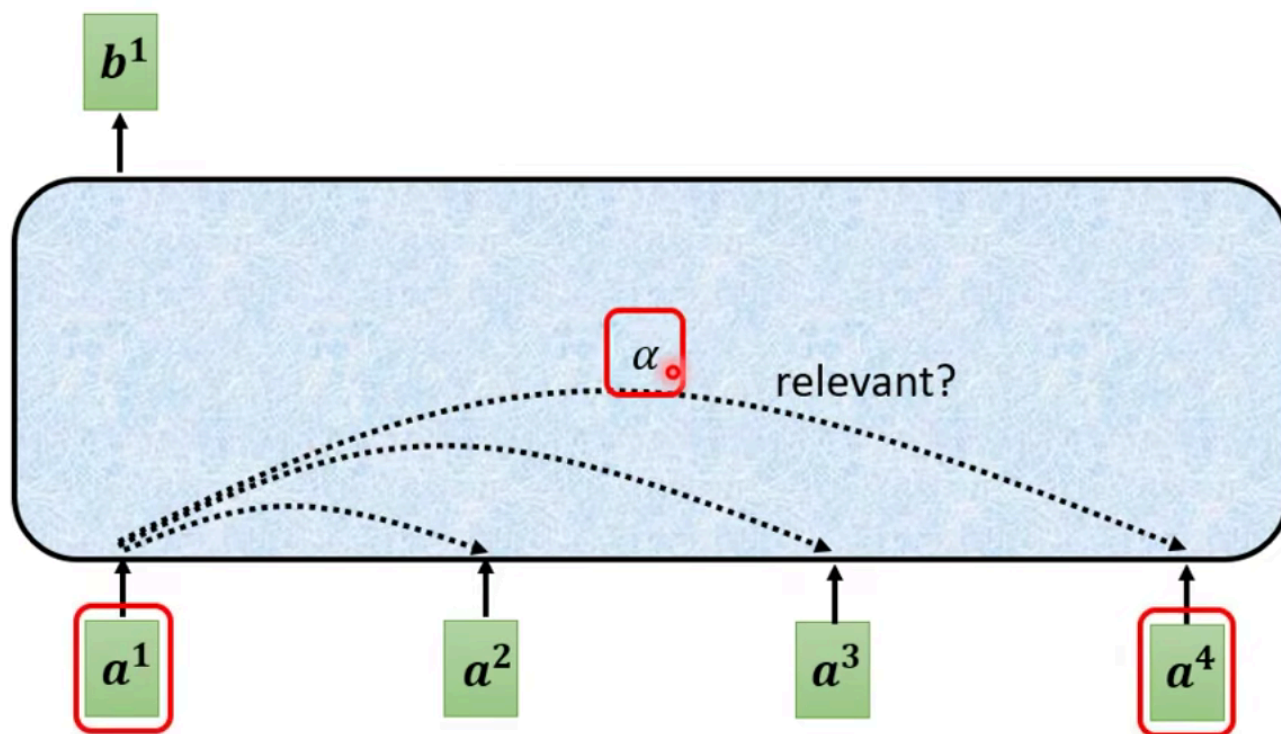
self-attention 处理的是整个序列的信息，得到考虑一整个序列的信息；FC(Fully Connected Network) 处理的是局部的信息。



self-attention可以不断叠加，与FC交替出现

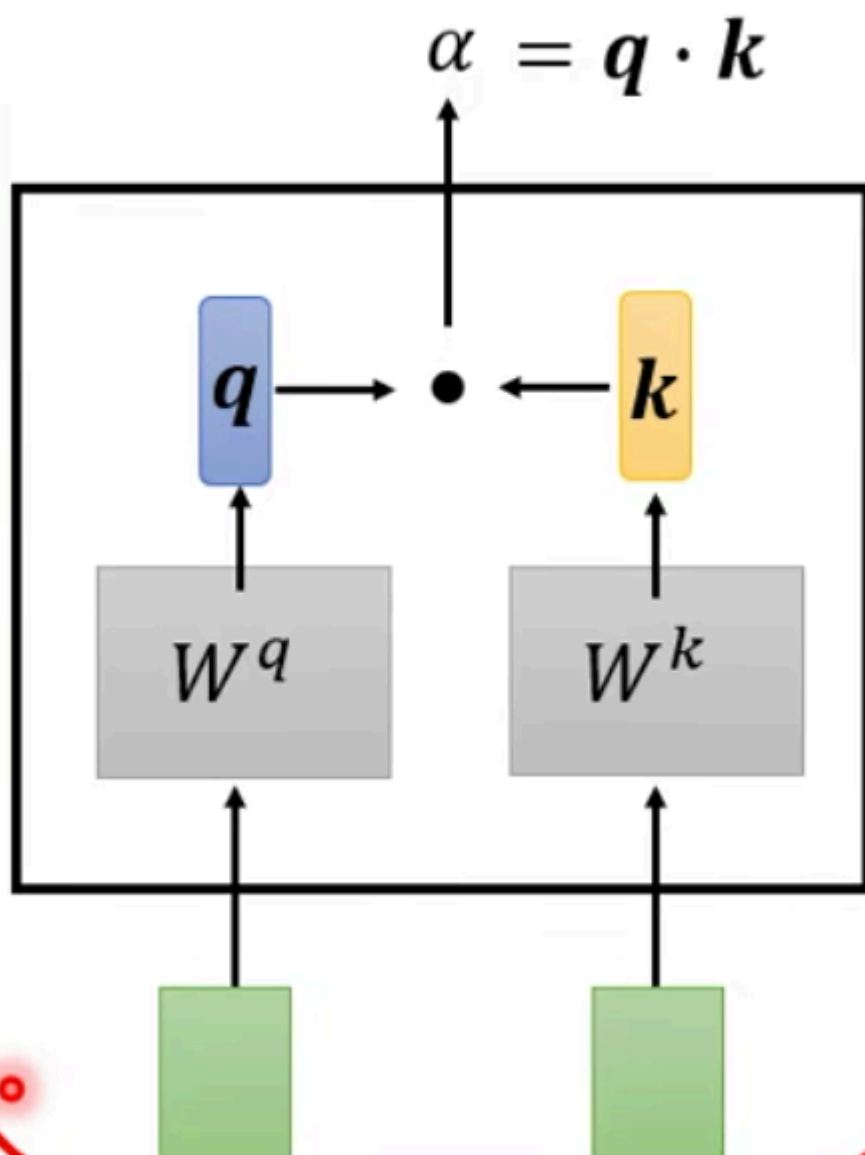
2.2 产生的步骤

1. 找出在序列中的相关的向量，用 α 代表两个向量的关联性



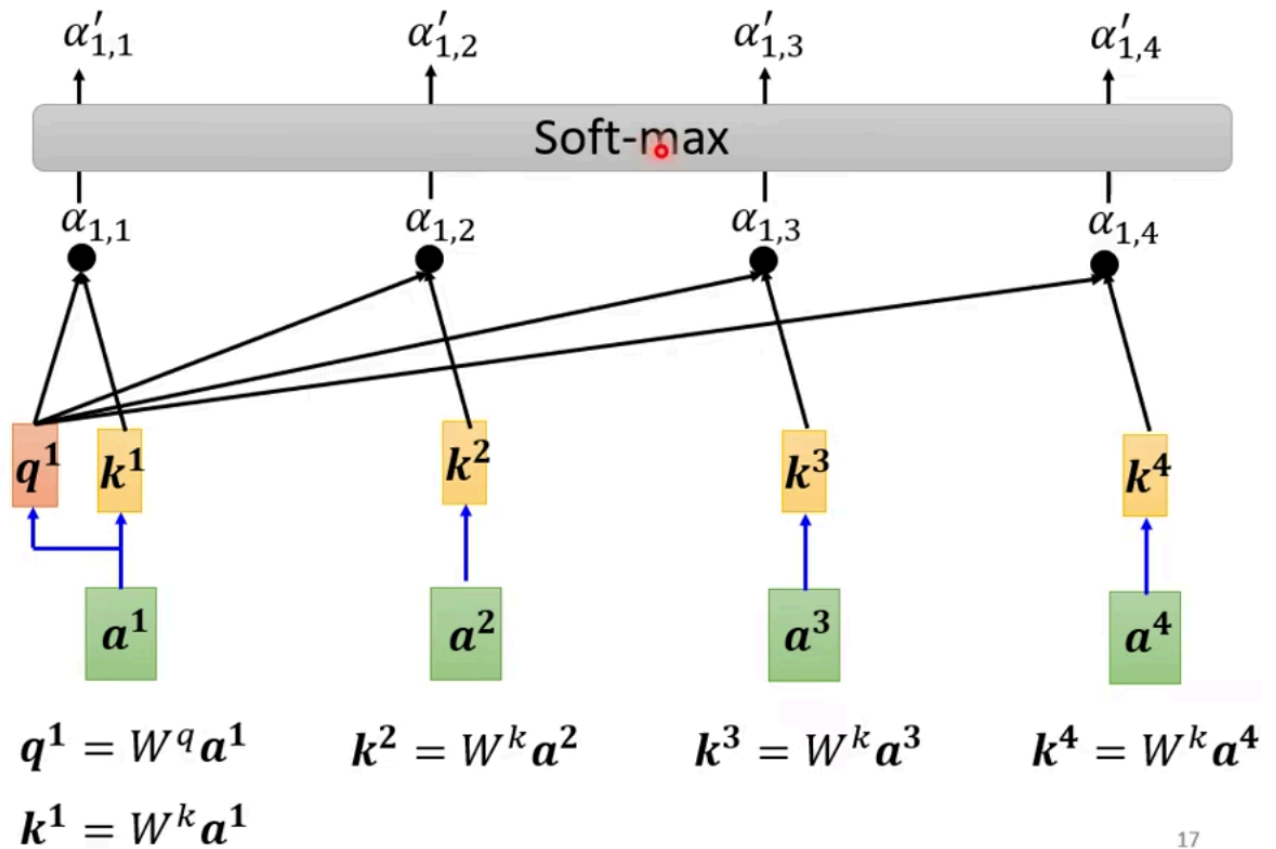
Find the relevant vectors in a sequence

Dot-product



Self-attention

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



17

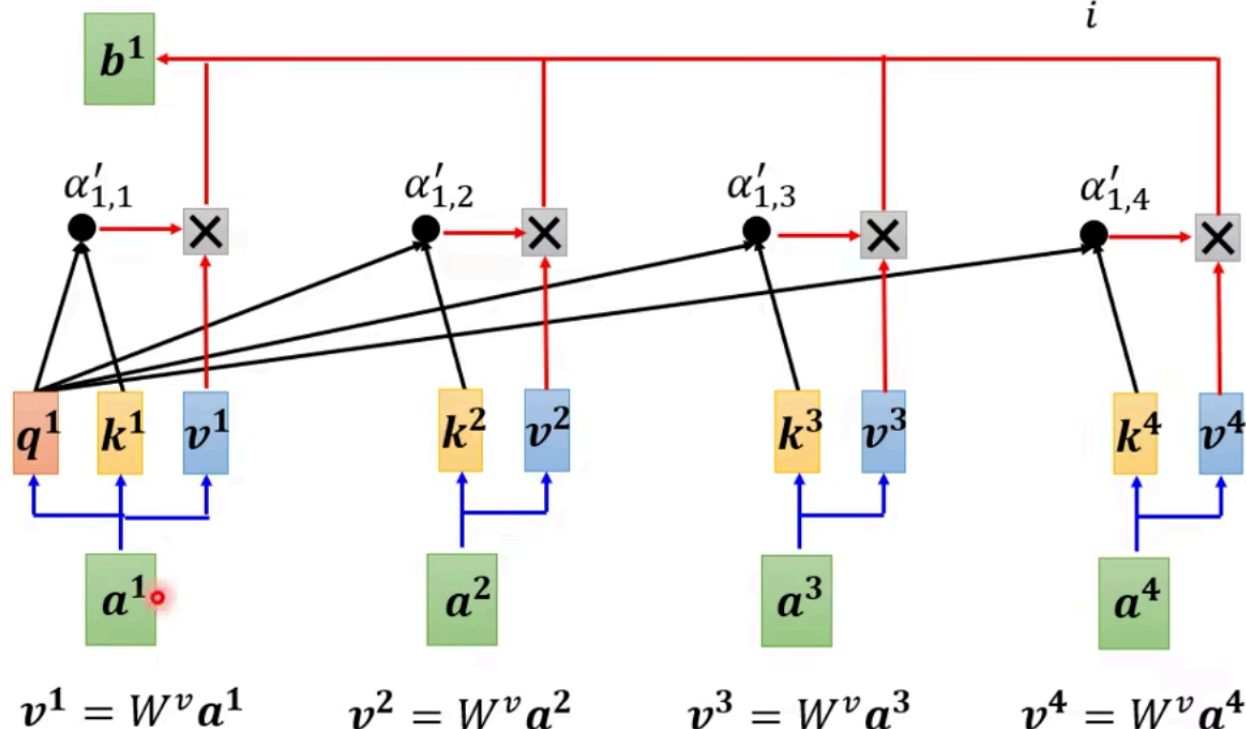
α^1 左乘 W^q , 剩余的阿尔法左乘 W^k , 分别得到 q^1 与 k^2, k^3, k^4 ……一般来说 α^1 也需要左乘 W^q , 得到 k^1 。将 q^1 与 k^1, k^2, k^3, k^4 ……点乘可以得到相关系数 $\alpha_{1,1} \alpha_{1,2} \alpha_{1,3}$ (可以叫做 attention score) …… 还需要通过Soft-max机制, 转换相关系数

2. 根据 $\alpha_{1,i}$ 抽取出序列中重要的信息

Self-attention

Extract information based on attention scores

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



将 a_i 左乘 W^v , 可以得到 v_i , v_i 与 $\alpha_{1,i}$ 相乘, 再将每一个乘积相加得到 b^1 , v_i 越大, 越接近抽取出来的结果 b^1

3. 矩阵表示

$$Q = W^q * I$$

$$K = W^k * I$$

$$V = W^v * I$$

$$Q = [q^1, q^2, q^3, q^4], K = [k^1, k^2, k^3, k^4],$$

$$V = [v^1, v^2, v^3, v^4], I = [\alpha^1, \alpha^2, \alpha^3, \alpha^4]$$

$$\begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1} \\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2} \\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3} \\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4} \end{bmatrix} = \begin{bmatrix} k_1^T \\ k_2^T \\ k_3^T \\ k_4^T \end{bmatrix} * [q_1 \quad q_2 \quad q_3 \quad q_4]$$

$$A = K^T * Q$$

$$A \rightarrow A'$$

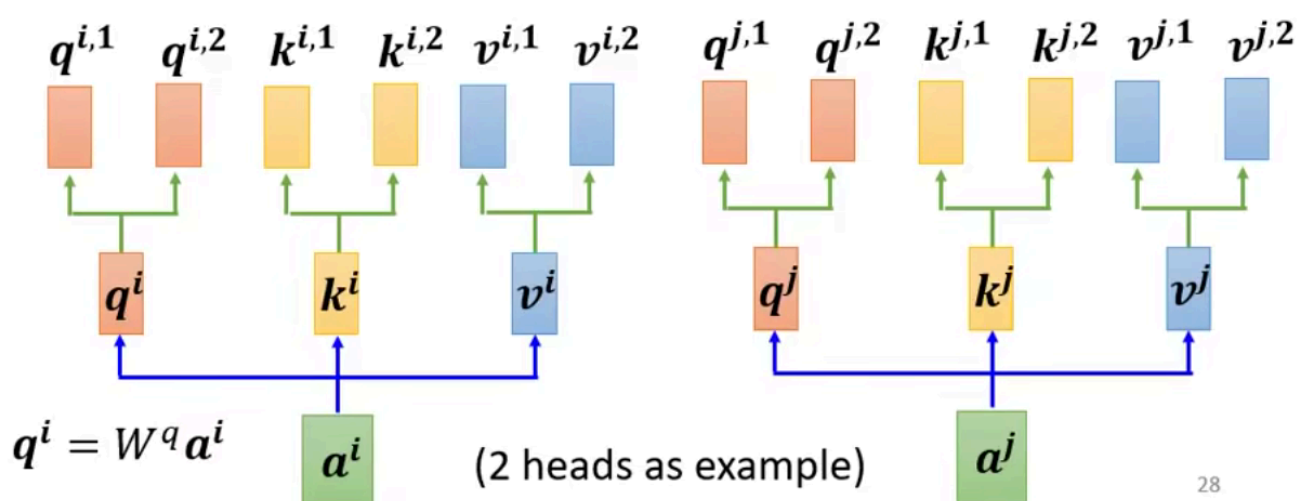
$$O = V * A'$$

W^q, W^k, W^v 需要通过学习找出来

4. Multi-head Self-attention

Multi-head Self-attention Different types of relevance

$$b^i = W^O \begin{bmatrix} b^{i,1} \\ b^{i,2} \end{bmatrix}$$

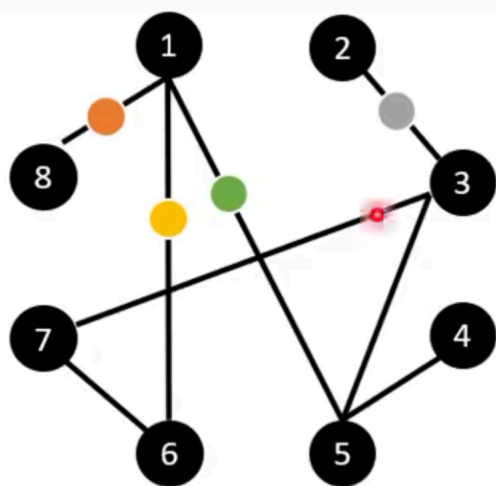


28

但是注意到并没有考虑位置，因此引入 e^i ，用 $e^i + \alpha^i$ 表示位置

5. 将self-attention应用于Graph

Self-attention for Graph



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

Attention Matrix

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

只需要计算有edge相连接的节点之间的相关系数

进一步的学习

- Long Range Arena: A Benchmark for Efficient Transformers
<https://arxiv.org/abs/2011.04006>
- Efficient Transformers: A Survey <https://arxiv.org/abs/2009.06732>