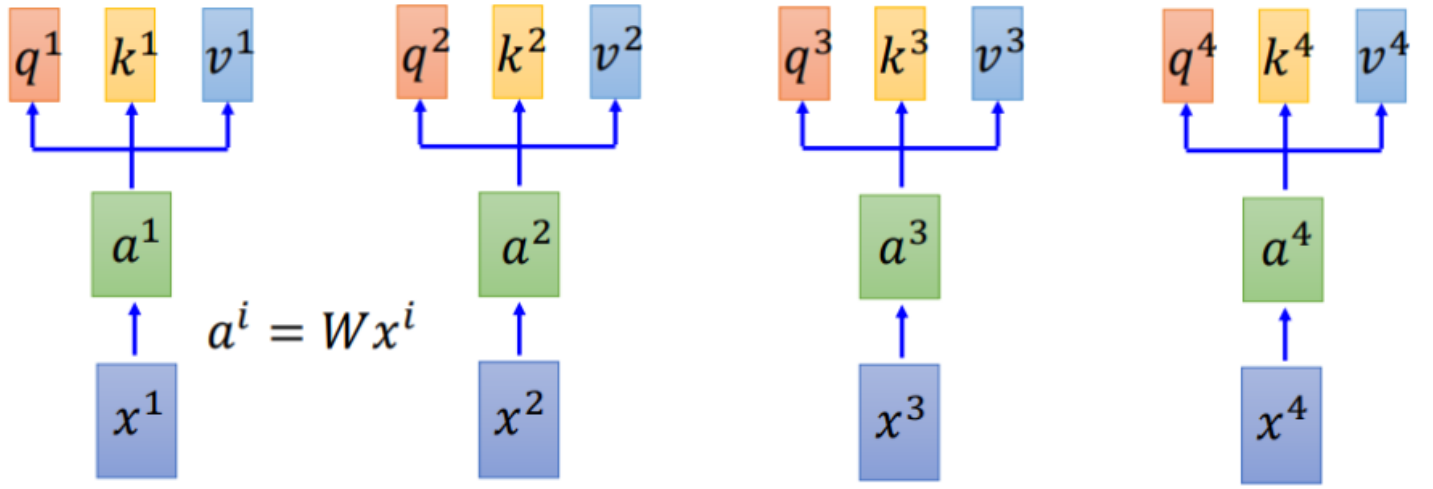
Self-attention Layer



q: query (to match others)

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

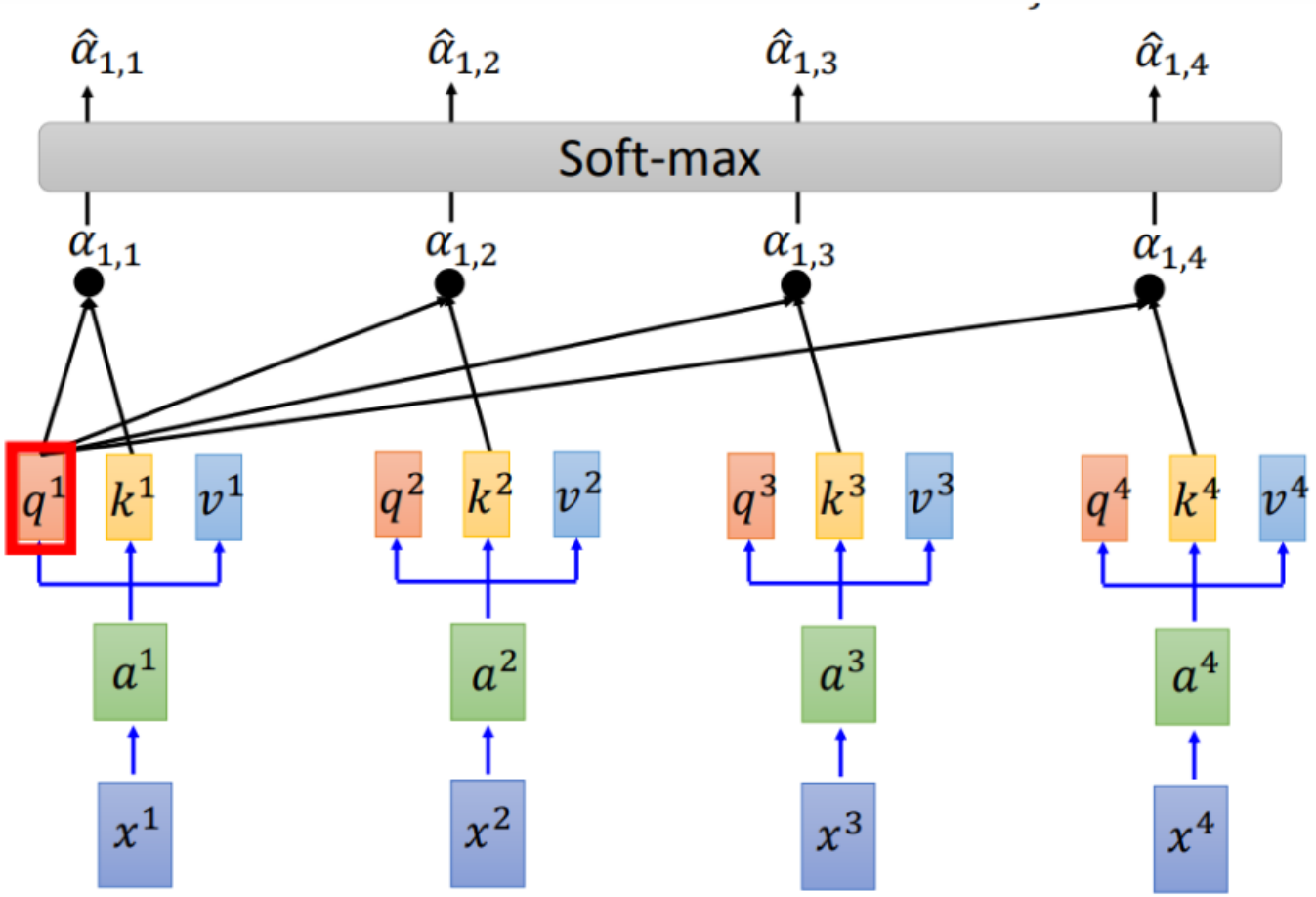
k: key (to be matched)

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

v: information to be extracted

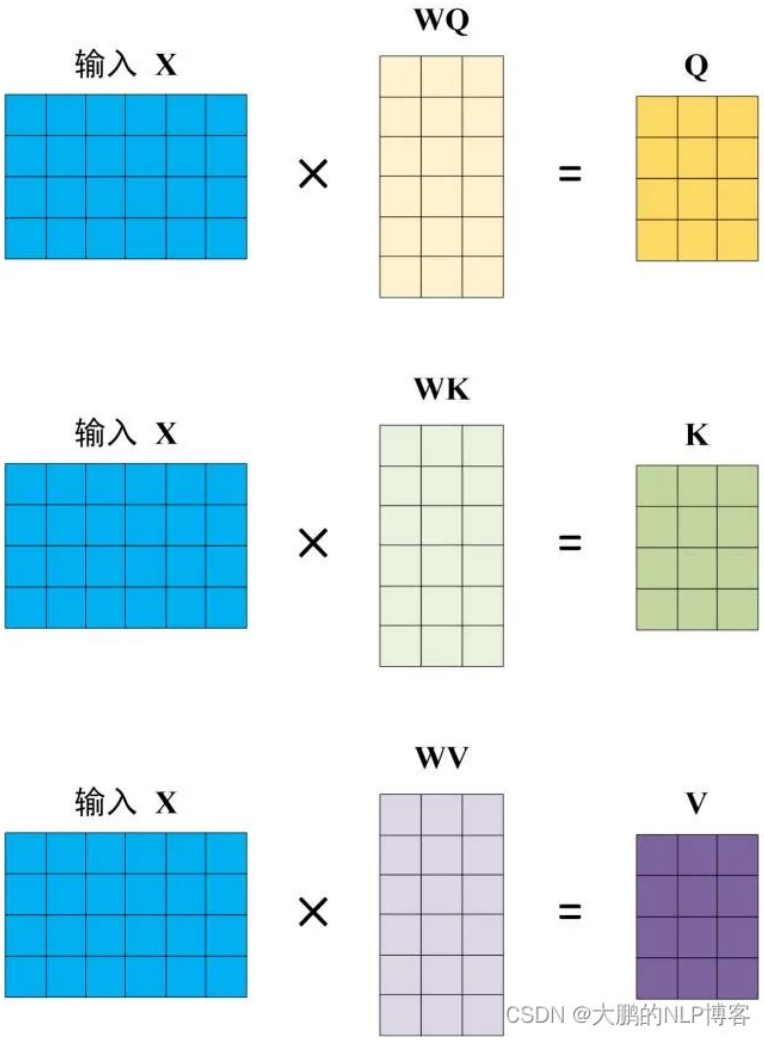
$$v^i = W^v a^i$$

Self-attention Layer



q对k做attentio $\alpha_{1,i} = \underbrace{q^1 \cdot k^i}_{\text{dot product}} / \sqrt{d}$

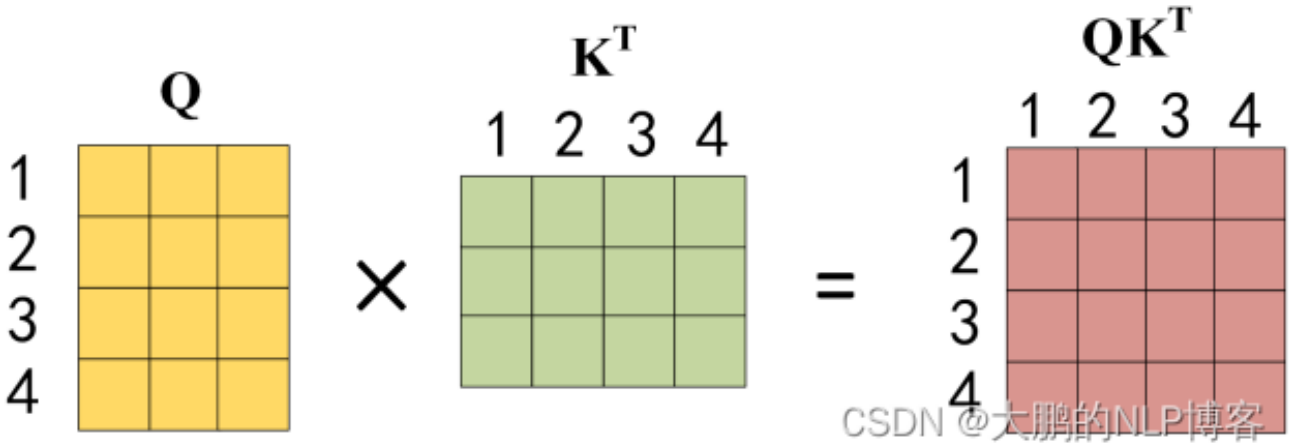
Attention



Self -Attention、Multi-Head Attention、Cross-Attention_cross attention-CSDN博客

Attention

Compute the weights of values



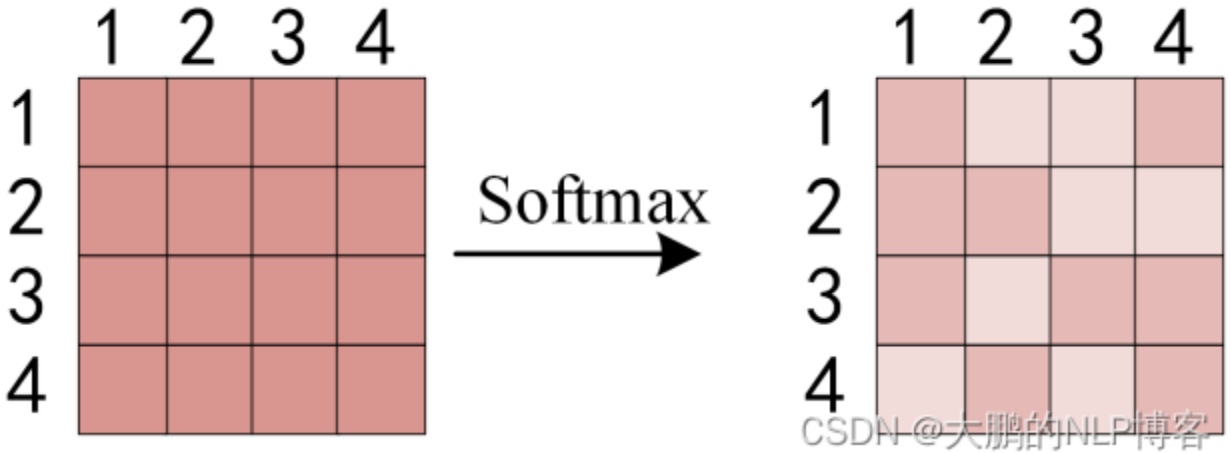
Note: In the formula, the inner product of each row vector of the matrix Q and K is calculated. In order to prevent the inner product from being too large, it is divided by $\sqrt{d_k}$, where d_k is the length of vector.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Self -Attention、Multi-Head Attention、Cross-Attention_cross attention-CSDN博客

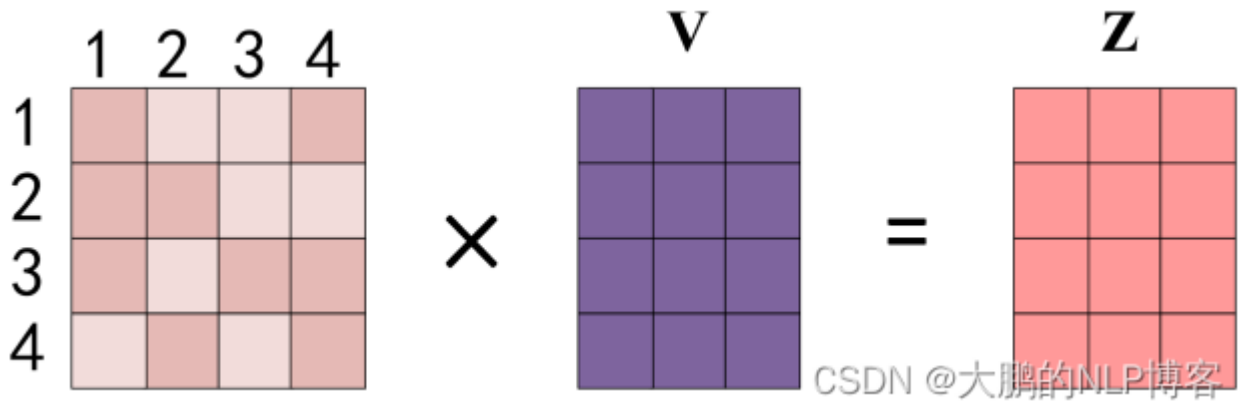
Attention

Normalize the weights by softmax



Attention

Output is the weighted-sum of V



Self-attention and Cross-attention

- Self-attention
  - Self-attention assigns different weights to inputs at different positions through learning, thereby better capturing long-range dependencies within the sequence.

- Cross-attention
  - Cross-attention enables the model to adjust the focus dynamically based on the content of another sequence while processing one sequence.