## 1. Attention

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

**为什么要除以$\sqrt{d_k}$**

随着$d_k$增大，$q$与 $k$的点积结果也在增大，会使得softmax梯度**变小**，甚至消失，所以$\frac{q*k}{\sqrt{d_k}}$

**优点**

- 一步到位获取全局与局部的联系，不会像RNN一样受到长期依赖的限制
- 每步的结果不依赖于上一步，可以做成并行的模式
- 参数量少，复杂度低 (在多数情况下)

**缺点**:

- 无法捕获位置信息，难以学习序列中的顺序关系（可以主动加入位置信息，例如BERT）

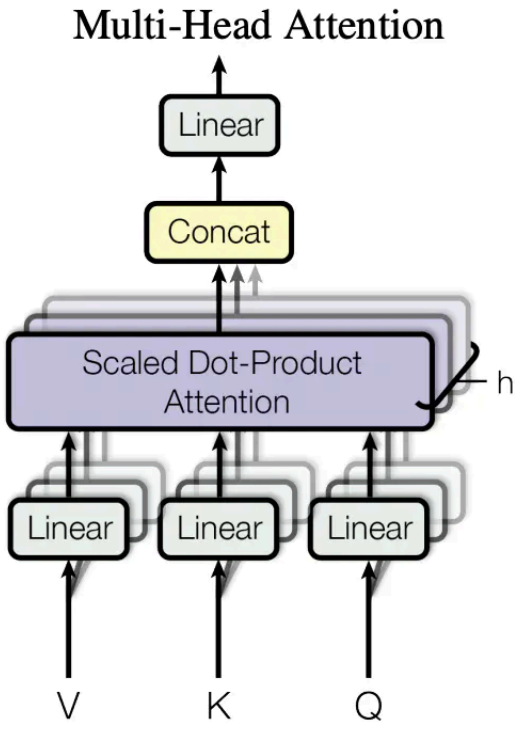
## 2. Self-Attention

该方法即Q,K,V都来自于同一个输入，其余计算过程，基本同上常用方法

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

## 3. Multi-Head Attention

> 8-head



$$Q_i = linear(Q).view(..8..)$$

$$K_i = linear(K).view(..8..)$$

$$V_i = linear(V).view(..8..)$$

$$head_i = Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i, i \in [1, 8]$$

$$MultiAttention(Q, K, V) = concat(head_1, head_2, \ldots, head_8)$$

**好处**

- Attention is all you need论文中讲模型分为多个头，形成多个子空间，每个头关注不同方面的信息
- 多头的本质是多个独立的attention计算，然后进行集成，具有一定防治过拟合的作用