

第六节 attention注意力机制

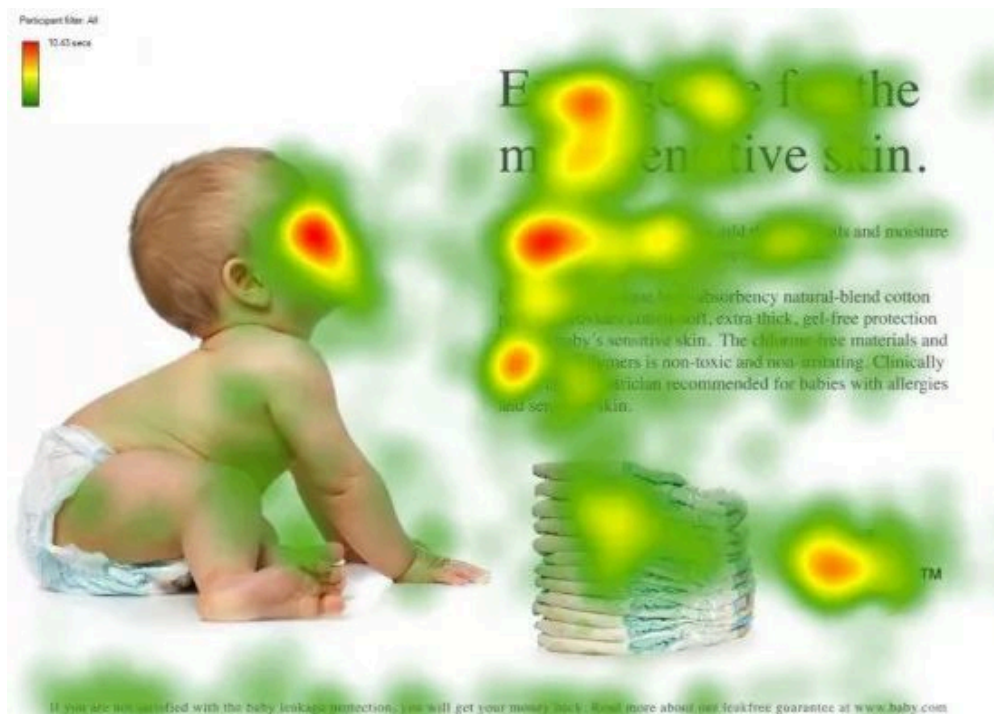
Attention is all you need

当看到一个事物的时候，你会注意什么？我们会发现，当我们人在关注一个事物的时候，我们往往会下意识关注一些重点，而会选择性地忽略掉一些无关紧要的细节或者背景，而这个就是attention注意力机制的由来。

大数据里面什么数据都有，包括重要的和不重要的。对于重要的数据，我们要使用；对于不重要的数据，我们要选择性地忽略或者起码要降低权重。这不仅在NLP是这样，CV、图等等所有的一切都可以套用这个道理。

但是，对于一个模型而言（CNN、LSTM），很难决定什么重要，什么不重要

由此，注意力机制诞生了（有人发现了如何去在深度学习的模型上做注意力）



红色的是科学家们发现，如果给你一张这个图，你眼睛的重点会聚焦在红色区域。也就是人--》看脸，文章看标题，段落看开头结尾

这些红色区域可能包含更多的信息，更重要的信息

也就是说，注意力机制就是我们会把我们的焦点聚焦在比较重要的事物上

注意力机制的核心算法

我（查询对象 Q ，这个查询对象可以是任何有意义的事物，包括被查询图片自身），这张图（被查询对象 V ）

我看这张图，第一眼，我就会去判断哪些东西对我而言更重要，哪些对我而言又更不重要（也就是去计算 V 相对于 Q 来说的重要程度）

重要程度计算，其实是不是就是相似度计算（比较接近），其实就是求内积（至于为什么可以，大家感兴趣的话可以去看一下矩阵运算的本质）

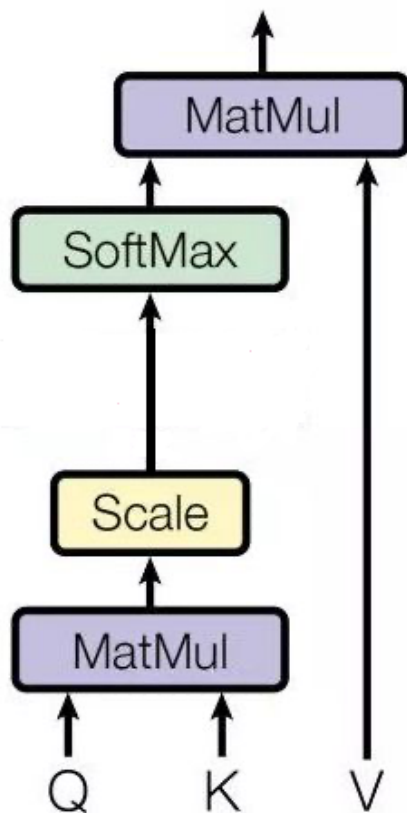
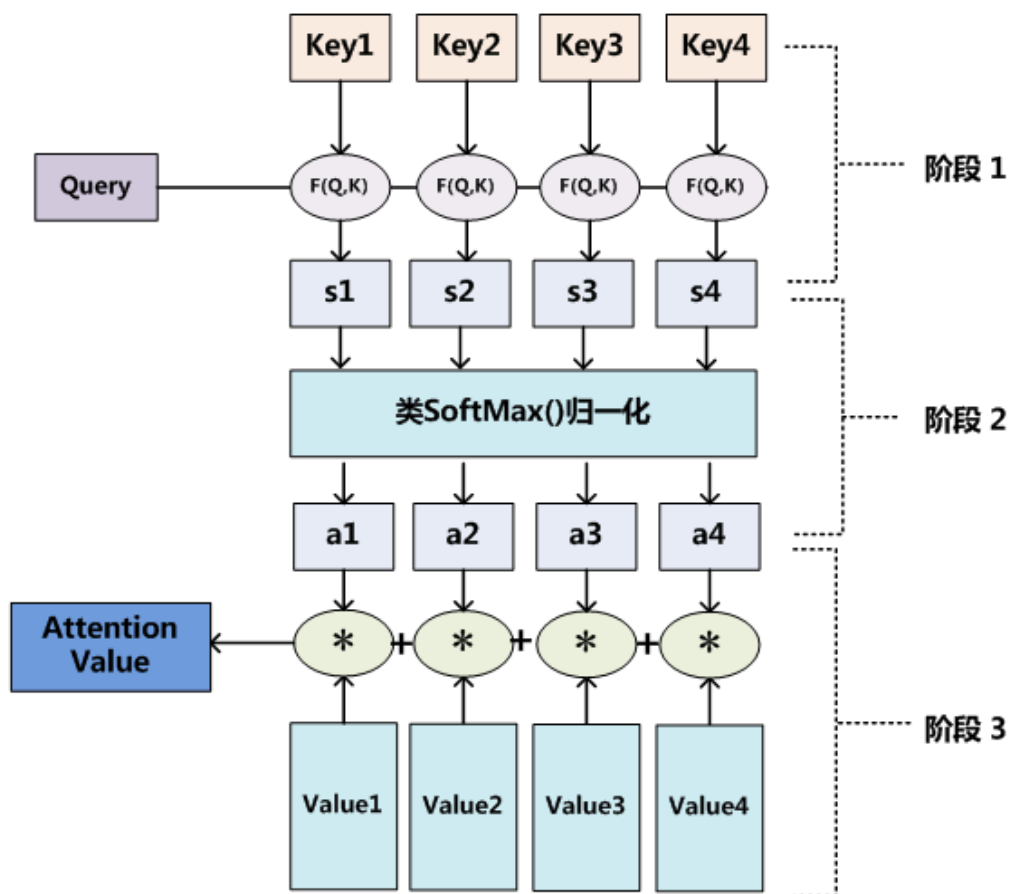
这里面我们引入一个矩阵 K ，一般而言 $K=V$ ，在 Transformer 里， $K \neq V$ 可不可以，可以的，但是 K 和 V 之间一定具有某种联系，这样的 QK 点乘才能指导 V 哪些重要，哪些不重要

$Q, K = k_1, k_2, \dots, k_n$ ，我们一般使用点乘的方式

通过点乘的方法计算 Q 和 K 里的每一个事物的相似度，就可以拿到 Q 和 k_1 的相似值 s_1 ， Q 和 k_2 的相似值 $s_2 \dots Q$ 和 k_n 的相似值 s_n

做一层 $\text{softmax}(s_1, s_2, \dots, s_n)$ 就可以得到概率 (a_1, a_2, \dots, a_n)

进而就可以找出哪个对 Q 而言更重要了



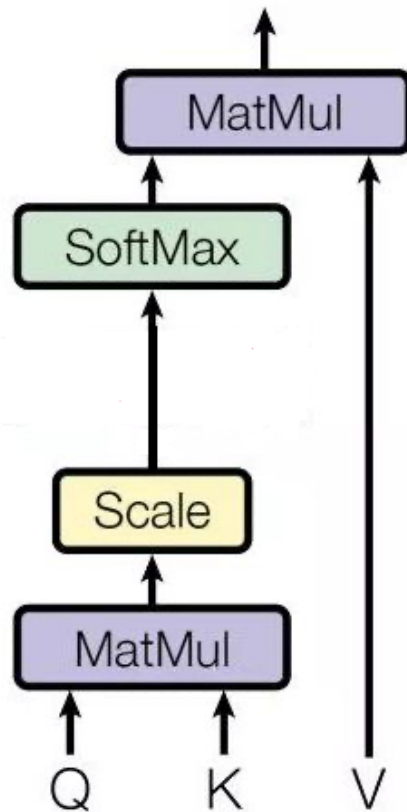
我们还得进行一个汇总，当你使用 Q 查询结束后，Q 已经失去了它的使用价值了，我们最终还是要拿到这张图片的，只不过现在的这张图片，它多了一些信息（多了于我而言更重要还是更不重要的信息在这里）

$$V = (v_1, v_2, \dots, v_n)$$

$$(a_1, a_2, \dots, a_n) * (v_1, v_2, \dots, v_n) = (a_1 * v_1 + a_2 * v_2 + \dots + a_n * v_n) = V'$$

这样的话，就得到了一个新的 V' ，这个新的 V' 就包含了，哪些更重要，哪些不重要的信息在里面，然后用 V' 代替 V

self-attention自注意力机制



QK 相乘求相似度，做一个 scale（缩放操作，保证得到的attention scores的方差在一个稳定的范围之内，未来做 softmax 的时候避免出现极端情况，也就是梯度爆炸、梯度弥散这些）

然后做 Softmax 得到概率

得到新的矩阵V'不仅包含了V的信息，然后这种表示还暗含了 Q 的信息（于 Q 而言，K 里面重要的信息），也就是说，挑出了 K 里面的关键点

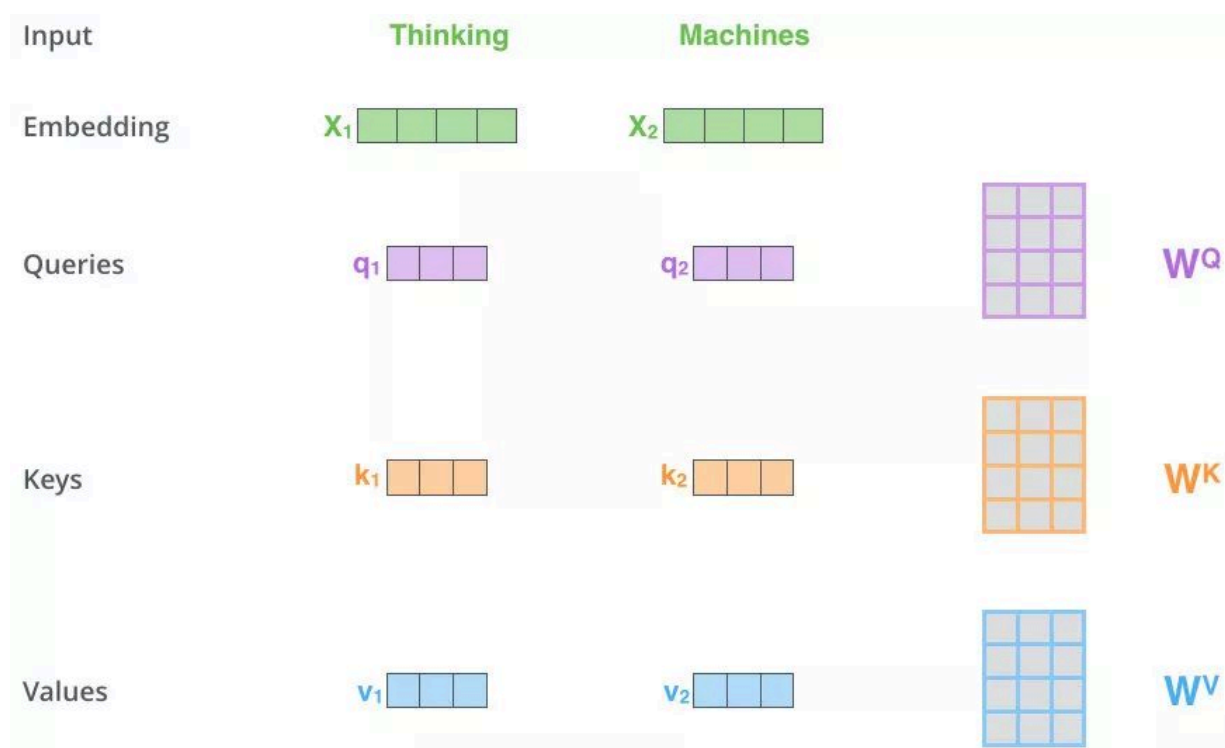
Self-Attention 的关键点再于， $K \approx V \approx Q$ 来源于同一个矩阵X，这三者是同源的

也就是通过 X 本身找到 X 里面的关键点

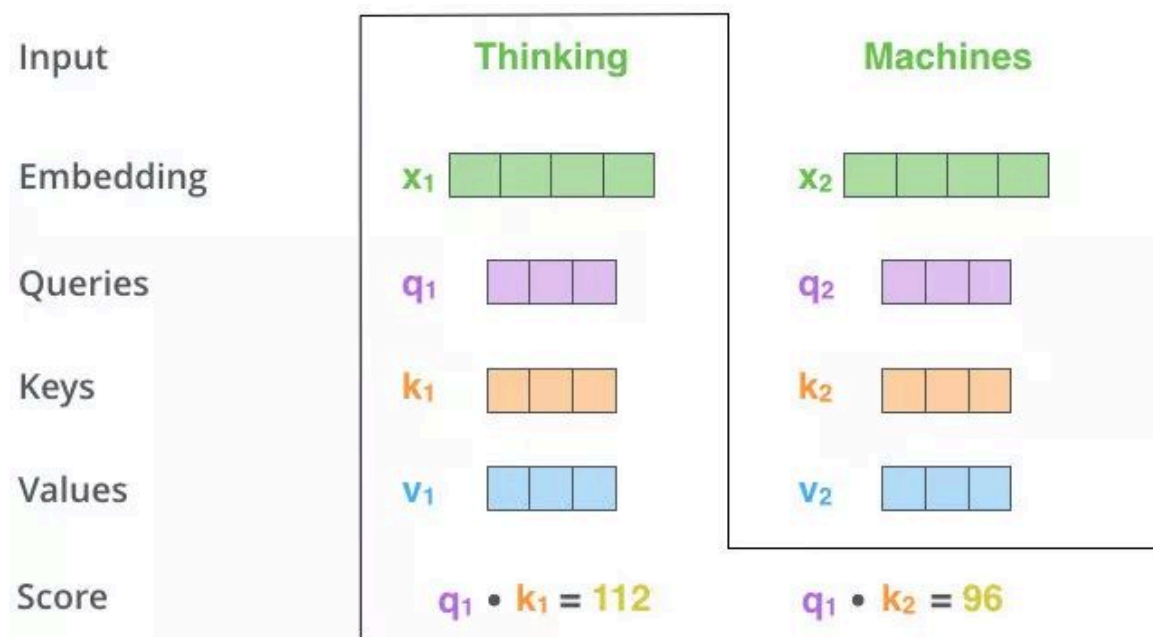
当然这里并不是 $K=V=Q=X$ ，而是用一些参数矩阵和X做运算得到Q、K、V，但是只是做了一些线性变换而已

接下来的步骤和注意力机制一模一样

1、Q、K、V的获取



2、Matmul



3、scale+softmax

Input

Embedding

Queries

Keys

Values

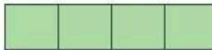
Score

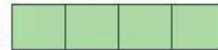
Divide by 8 ($\sqrt{d_k}$)

Softmax

Thinking

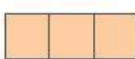
Machines

x_1 

x_2 

q_1 

q_2 

k_1 

k_2 

v_1 

v_2 

$$q_1 \cdot k_1 = 112$$

$$q_1 \cdot k_2 = 96$$

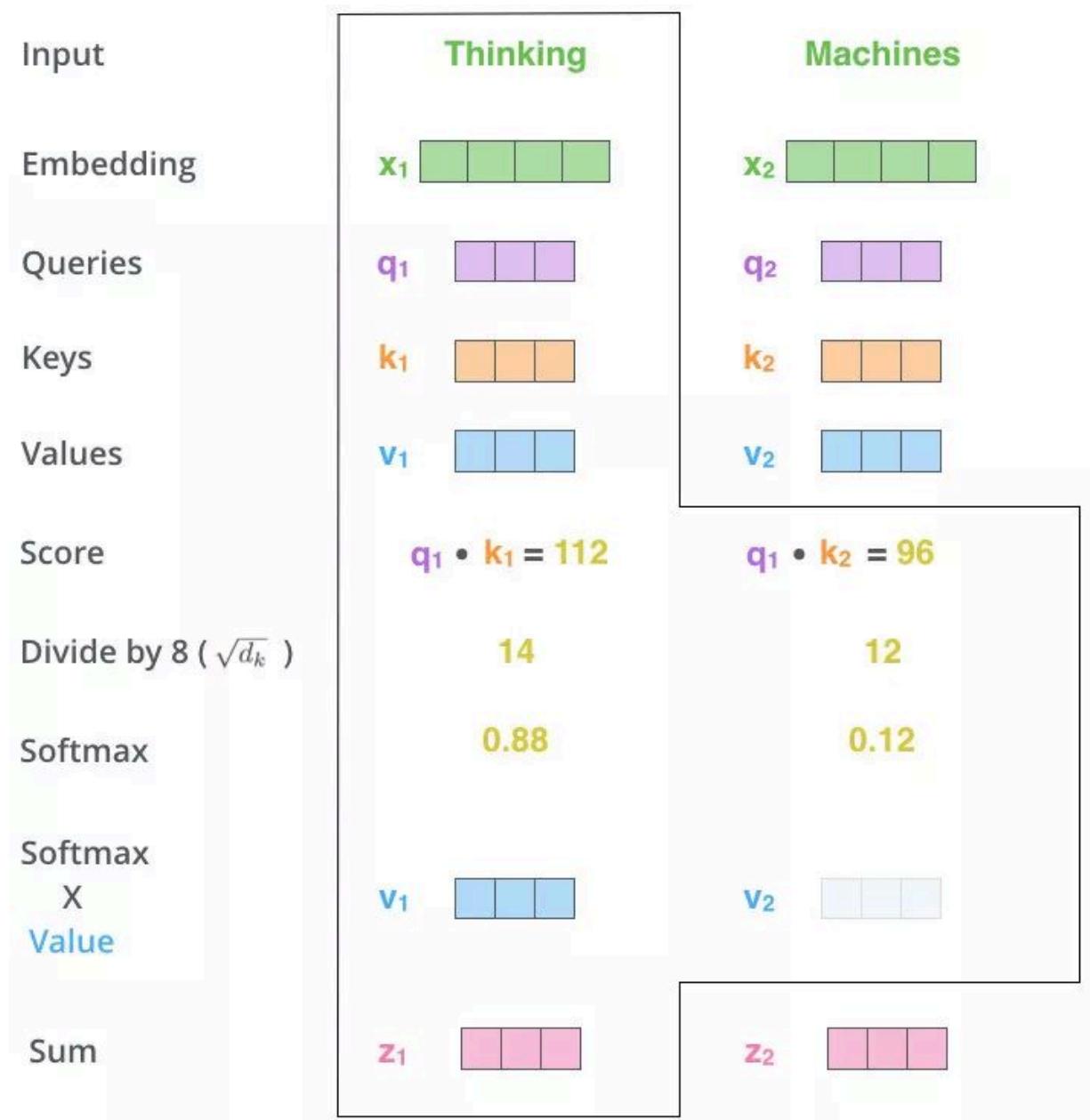
14

12

0.88

0.12

4、Matmul

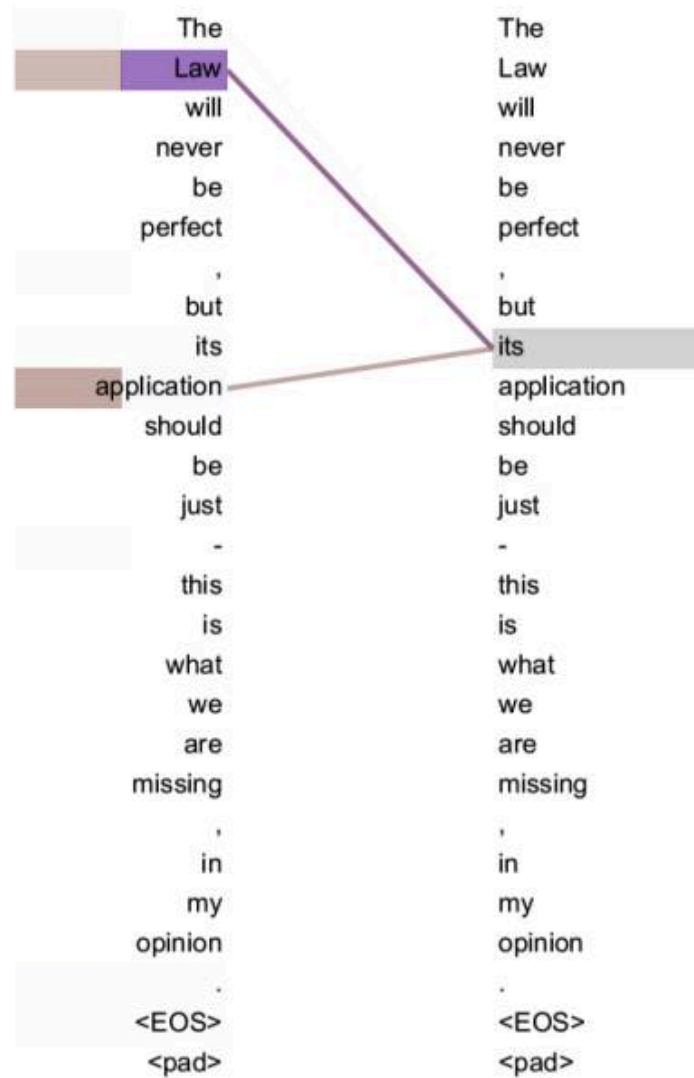


z_1 表示的就是 thinking 的新的向量表示

对于 thinking, 初始词向量为 x_1

现在我通过 thinking machines 这句话去查询这句话里的每一个单词和 thinking 之间的相似度

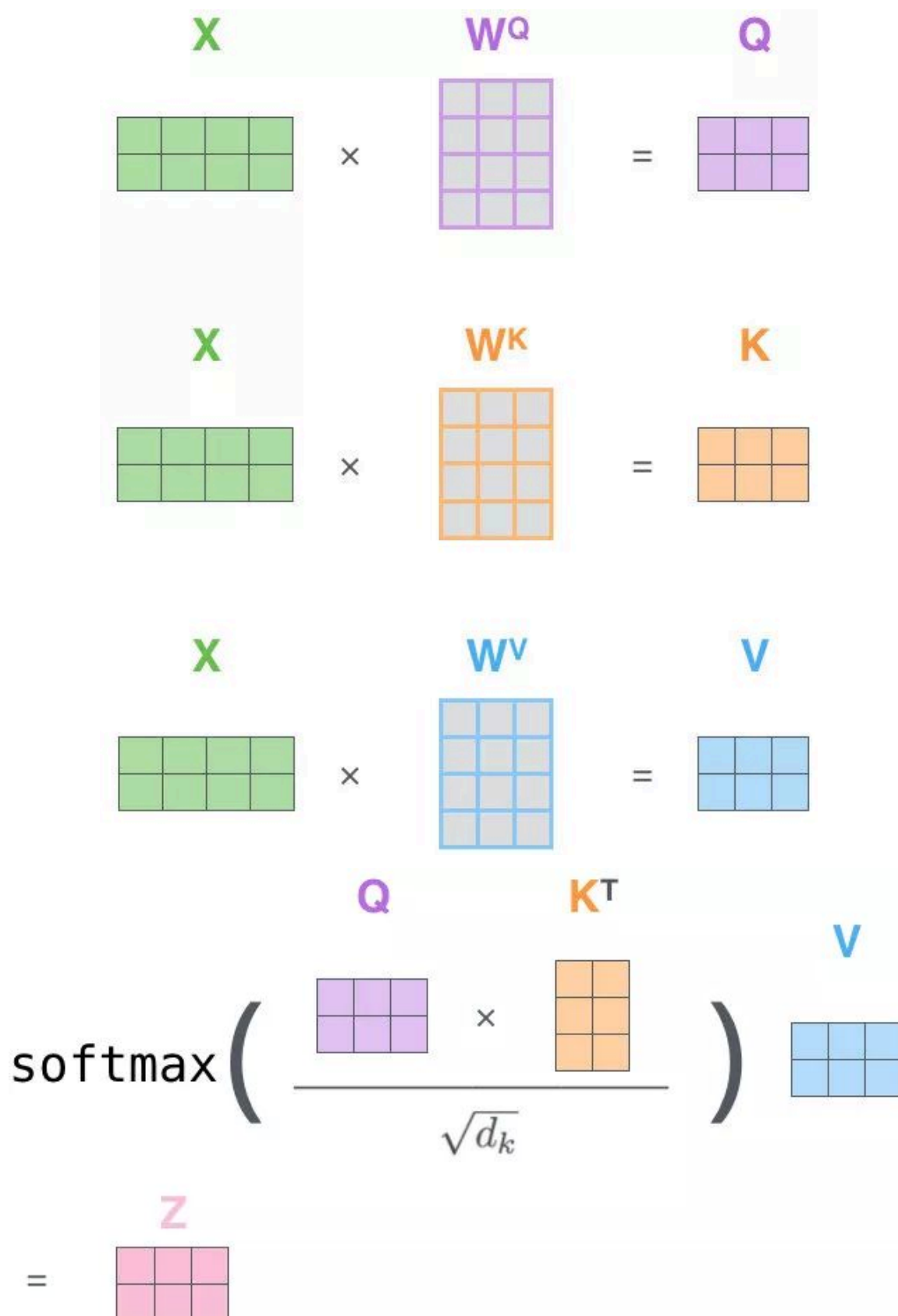
新的 z_1 依然是 thinking 的词向量表示, 只不过这个词向量的表示蕴含了 thinking machines 这句话对于 thinking 而言哪个更重要的信息



不做注意力，its 的词向量就是单纯的 its，没有任何附加信息

在做了attention的操作之后，its 有了 law 这层意思（当然其实是包含了一整句话甚至是一整段文章的信息，一般是一句话，不然计算复杂度太高了）

自注意力机制（矩阵）





Masked self-attention (掩码自注意力机制)

为什么要做这个改进：在做生成模型去生成单词的时候，我们是一个一个生成的

当我们做生成任务的时候，我们也想对这个生成的单词做注意力计算，但是，生成的句子是一个一个单词生成的，也就是说后面的单词其实对于前面的单词来说其实是不可见的

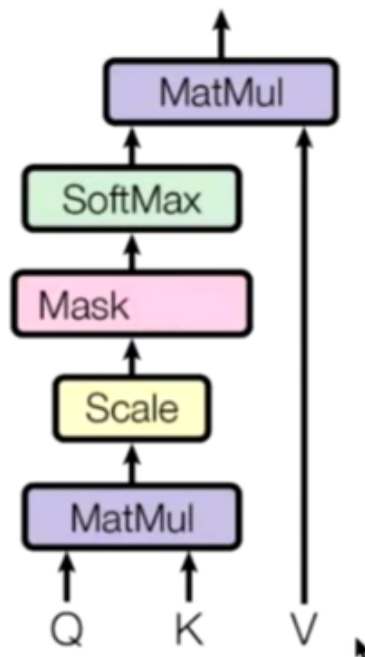
I have a dream

1. I 第一次注意力计算，只有 I
2. I have 第二次，只有 I 和 have
3. I have a
4. I have a dream
5. I have a dream

所以为了适应生成任务的需要，掩码自注意力机制应运而生

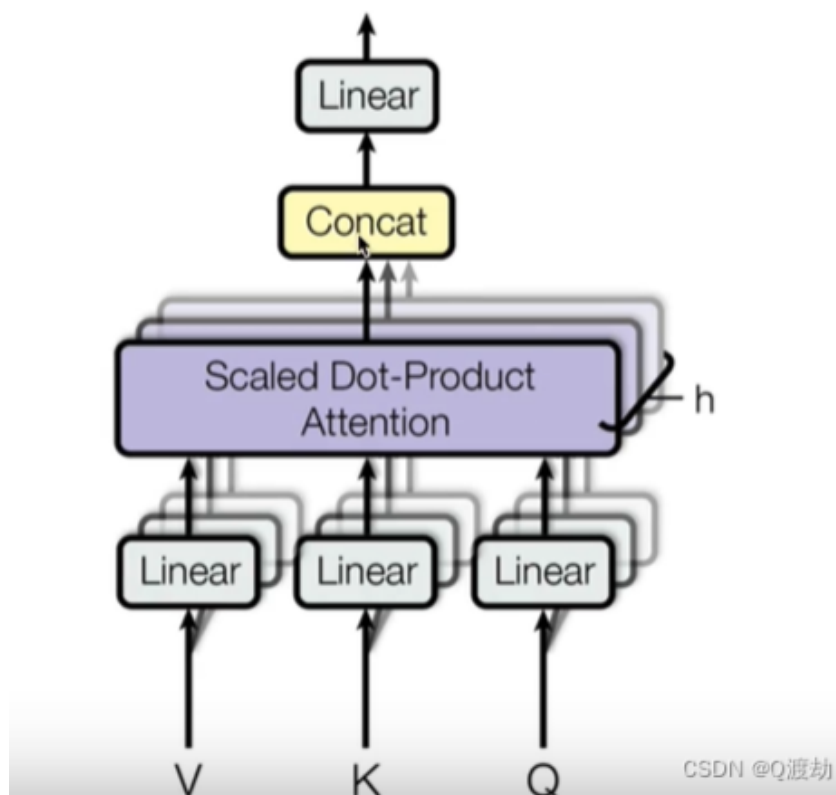
	I	have	a	dream
I				
have				
a				
dream				

	I	have	a	dream
I	1			
have	0.4	0.6		
a	0.1	0.1	0.8	
dream	0.2	0.3	0.1	0.4



其实这里很简单，就是对于前面的单词来说，后面的单词不参与运算，只和前面的单词运算，这样就能保证词向量更符合现实的分布。

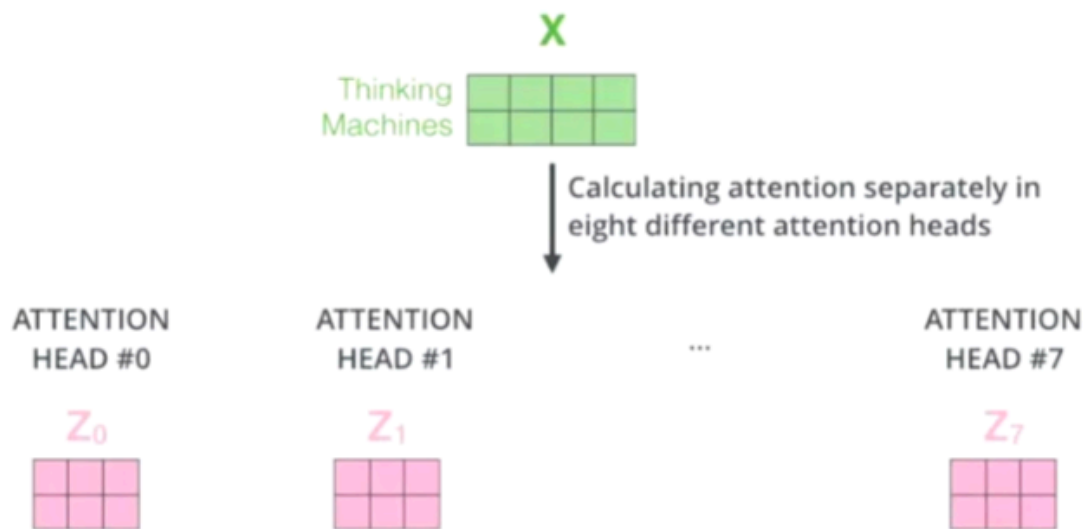
Multi-head attention（多头注意力机制）



我们在上面的masked self-attention中，通过一系列操作将原始输入X变换成了输出Z，这个Z包含了对X自身而言哪些东西更重要的信息。

而现在，再经过多头注意力，我们可以进一步上面的操作，得到Z'，这个Z'又是对比Z更好地描述了对X而言哪些东西更重要。

什么是多头



对于X，我们不是直接拿X去得到最终的Z，而是把X分成了h块（这里面一般h=8，也就是我们经常使用的是8头注意力），得到z0到z7

1) Concatenate all the attention heads



2) Multiply with a weight matrix W^O that was trained jointly with the model

X

3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN



然后把z0到z7拼接起来（concat操作），再做一次线性变换（改变维度）得到Z

多头的作用

机器学习的本质是什么？

$$y = \sigma(wx + b)$$

在做一件什么事情呢？其实就是非线性变换+线性变换（或者说线性变换其实是非线性变换的一种特殊情况），其实也就是信息在空间（低维度、高维度都有可能）当中的变换

非线性变换的本质就是改变空间上的位置坐标，任何一个点都能在维度空间上找到，也就是通过多次复杂的非线性变换，我们可以让一个不合理的点（位置不合理）变得合理

这就是词向量的本质，也就是说，我们从独热编码到word2vec到ELMO，再到现在的各种attention，都是在找寻一种更合适更好的方法得到这个合理的位置，也就是得到的更好的词向量

所以这里多头无非就是一种效果更好的能找到这个合理位置的方法，也就是通过多头分割，把X分到了8个不同的位置，从这8个不同的位置出发去寻找最终合理的位置。其实打个比方就是，我们玩捉迷藏要找一个人，假如只有我们一个人是不是很难找，假如我们有八个人从不同的方向去找是不是就简单很多，所以多头其实就是这个道理

那为什么不是分成100、1000头呢？岂不是越多越好？其实并不是这样，因为我们最终的结果是要做一个拼接的，假如说有100个头，里面有80个错误的，那么最终汇总的时候就会掺杂很多错误的信息，从而导致不准确，8头是一个经过多次实验得到的比较合适的一个数量。