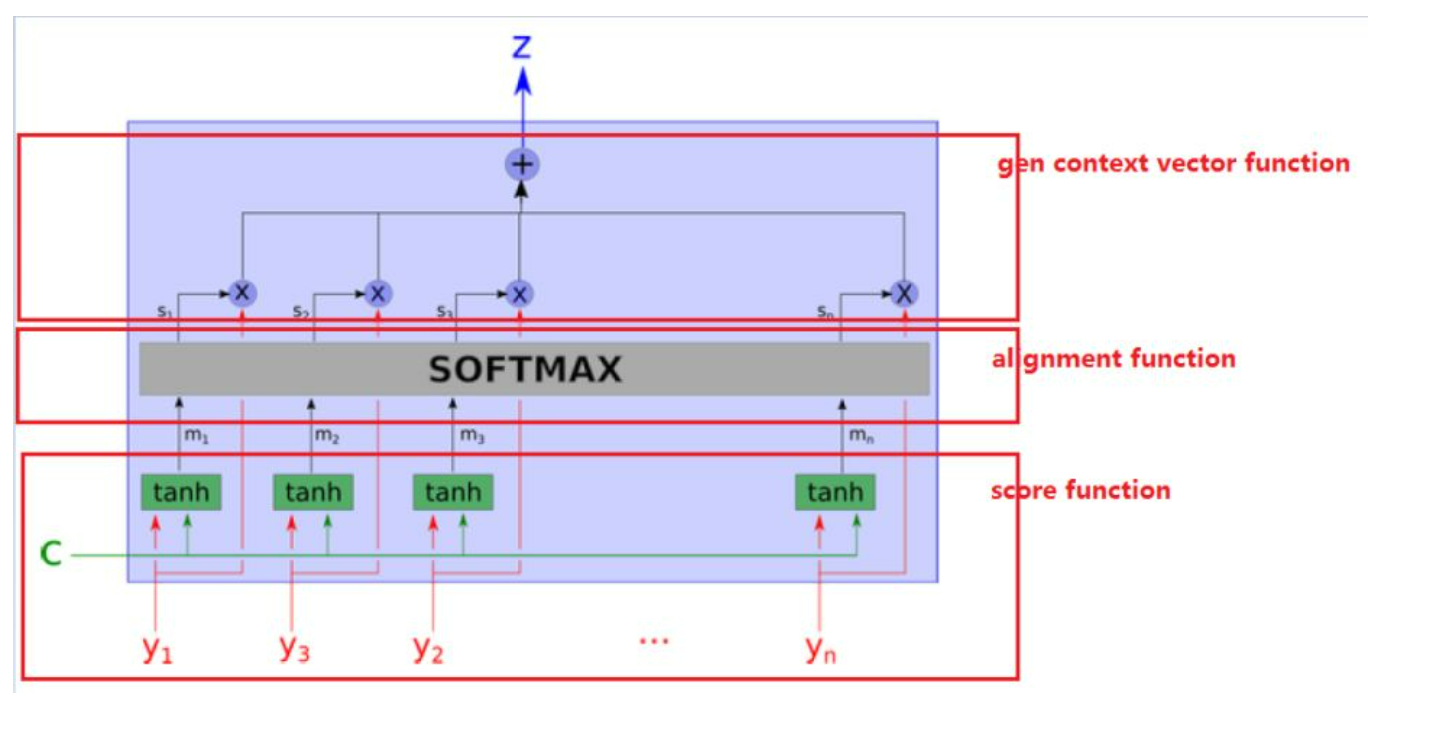
## Attention

1. 什么是Attention：输出z是输入的“**加权求和**”，权重是根据给定上下文c时，每个输入y的相关性来选择的。两个角度来看：

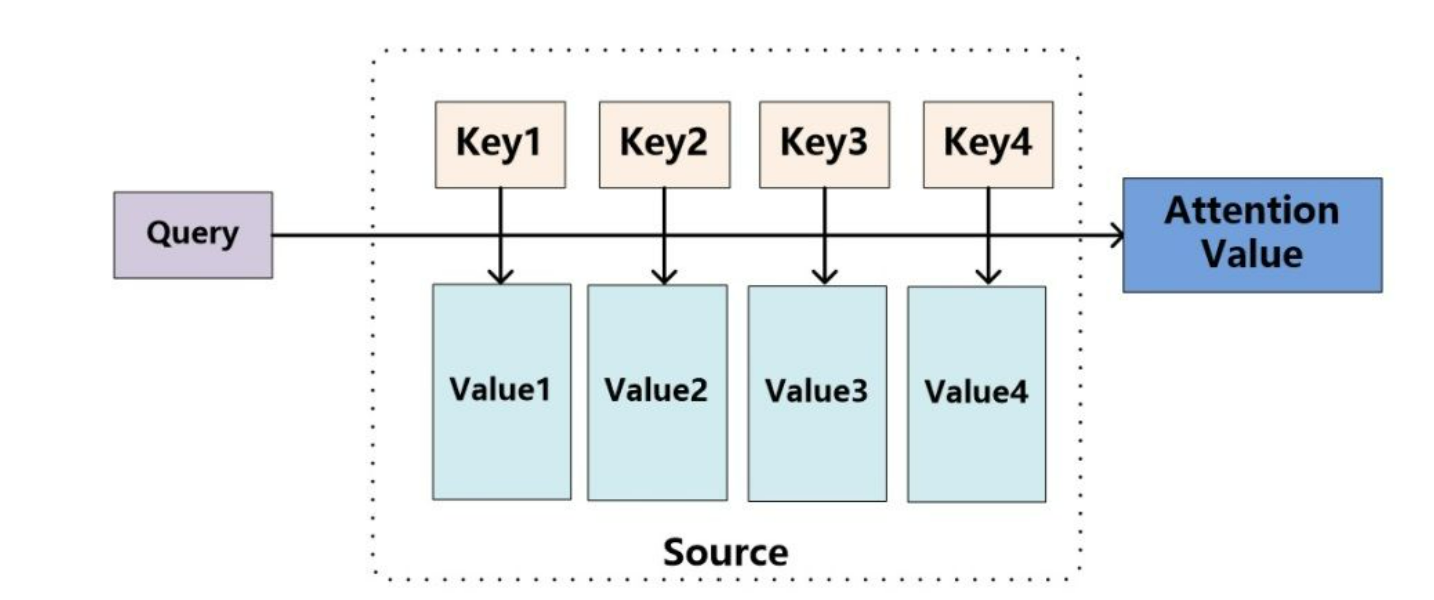
(1) 经典的 Bahdanau attention：

* score function：度量环境向量与当前输入向量的相似性，找到当前环境下，应该 focus 哪些输入信息
* alignment function：计算注意力权重，通常都使用softmax进行归一化
* generate context vector function：根据注意力权重，得到输出向量



(2) QKV模型：

* address memory(score function）:
* normalize(alignment function）：
* read content(gen context vector function）：



1. 各种不同的Attention
2. hard / soft attention：（soft attention更常用）

* hard attention 是一个随机采样，采样集合是输入向量的集合，采样的概率分布是alignment function 产出的 attention weight。因此，hard attention 的输出是某一个特定的输入向量。
* soft attention 是一个带权求和的过程，求和集合是输入向量的集合，对应权重是alignment function产出的attention weight。（后文提及的所有 attention 都在这个范畴）

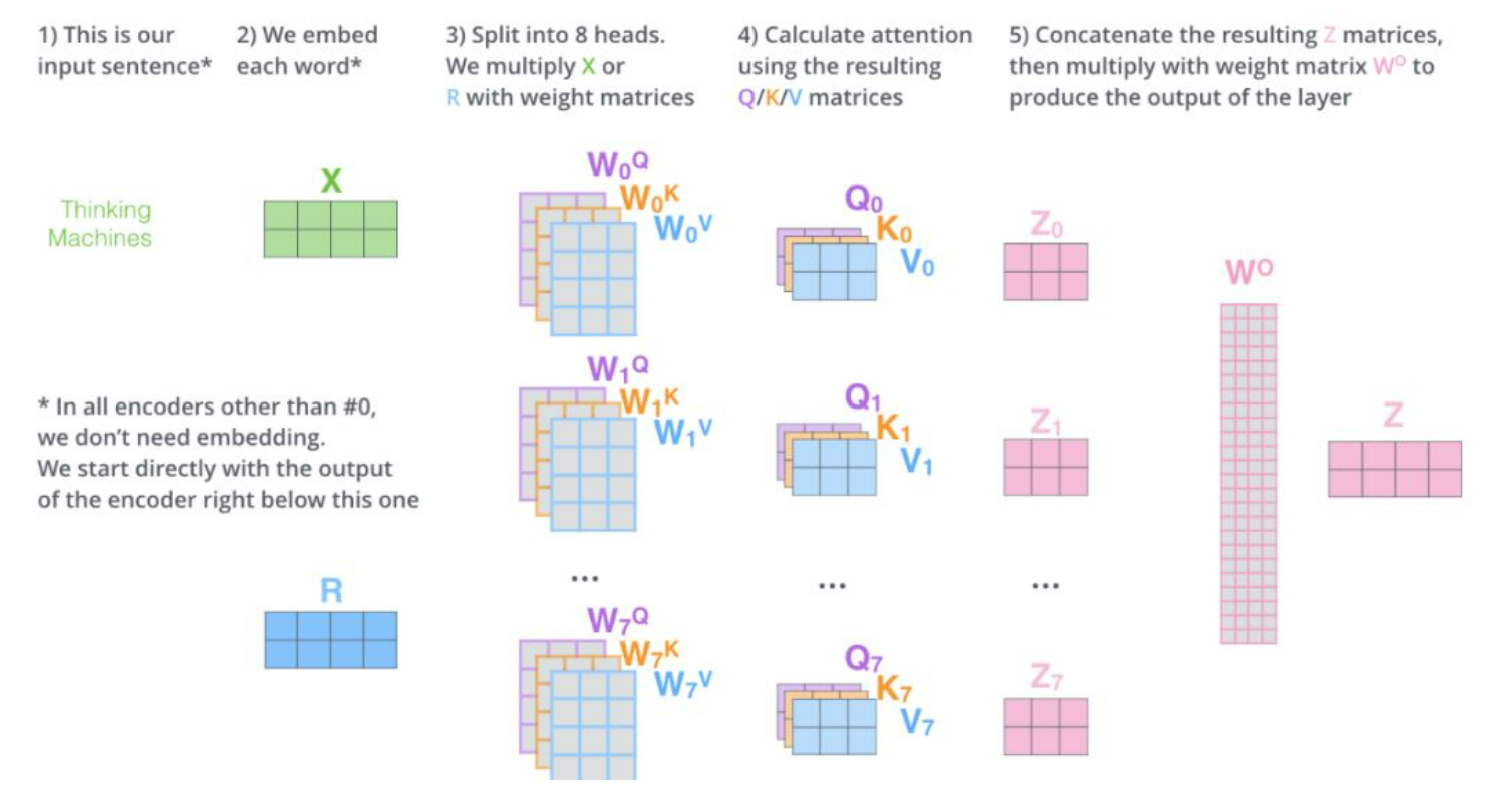
1. Self-Attention & Multi-head Attention

* 为什么用：已有的神经网络架构的优缺点如下，Self-Attention综合了优点

|  |  |  |
| --- | --- | --- |
|  | 优点 | 缺点 |
| RNN | 可以提取任意长序列的信息 | 无法并行 |
| CNN | 并行程度高 | 窗口固定，无法顾全全局信息 |



* 什么是自注意力：Q=K=V，下图是multi-head attention完整过程



* Transformer中，有三处使用了Attention

1. Encoder self-attention：Encoder阶段捕获当前word和其他输入词的关联；
2. MaskedDecoder self-attention：Decoder阶段捕获当前word与已经看到的解码词之间的关联，从矩阵上直观来看就是一个带有mask 的三角矩阵（看不见word之后的词）；
3. Encoder-Decoder Attention：就是将Decoder和Encoder输入建立联系，和之前那些普通Attention一样；

