**3/11 Sequence as input**

**影片1 自注意力机制 (Self-attention) (上)**

当输入的***向量是多个***且***向量长度会改变***的时候，称为“Sequence as input”。

例子1：自然语句输入。给定一个句子，常见的将句子表示在空间中的方法是one-hot编码以及Word Embedding，前者的想法是将每一个单词用一个one-hot编码表出，而后者的想法是将词汇表示为空间中的某个点。

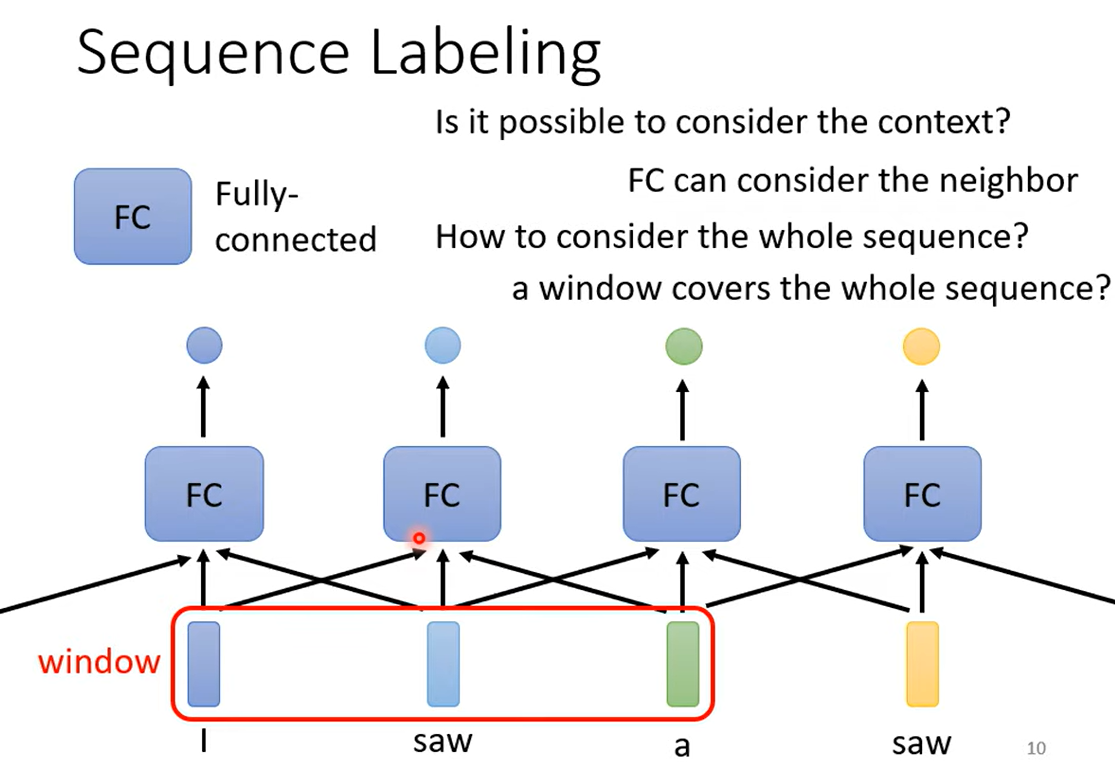
例子2：语音输出。通常我们会把一段语音分为多个window，每个window是固定的时长，由此形成一堆向量。

例子3：图结构中的节点。

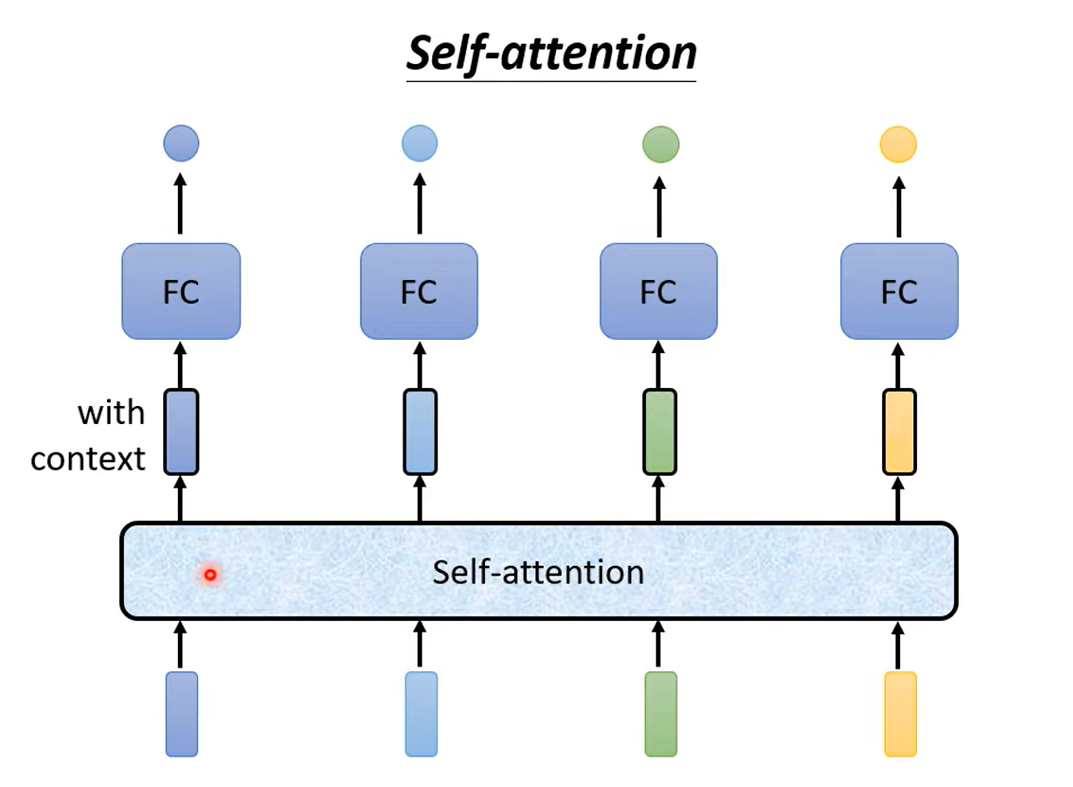
可能的输出：

1. 每个输入的向量都有一个对应的输出label（ex，为语句中的词性做判别）
2. 只需要输出一个label（ex，情感分析，即输入一段文字，得到对应的情感）
3. 机器自己决定输出的label数量（即seq2seq问题，ex翻译）

***类别1（每个输入对应输出一个label）—— Sequence Labeling***



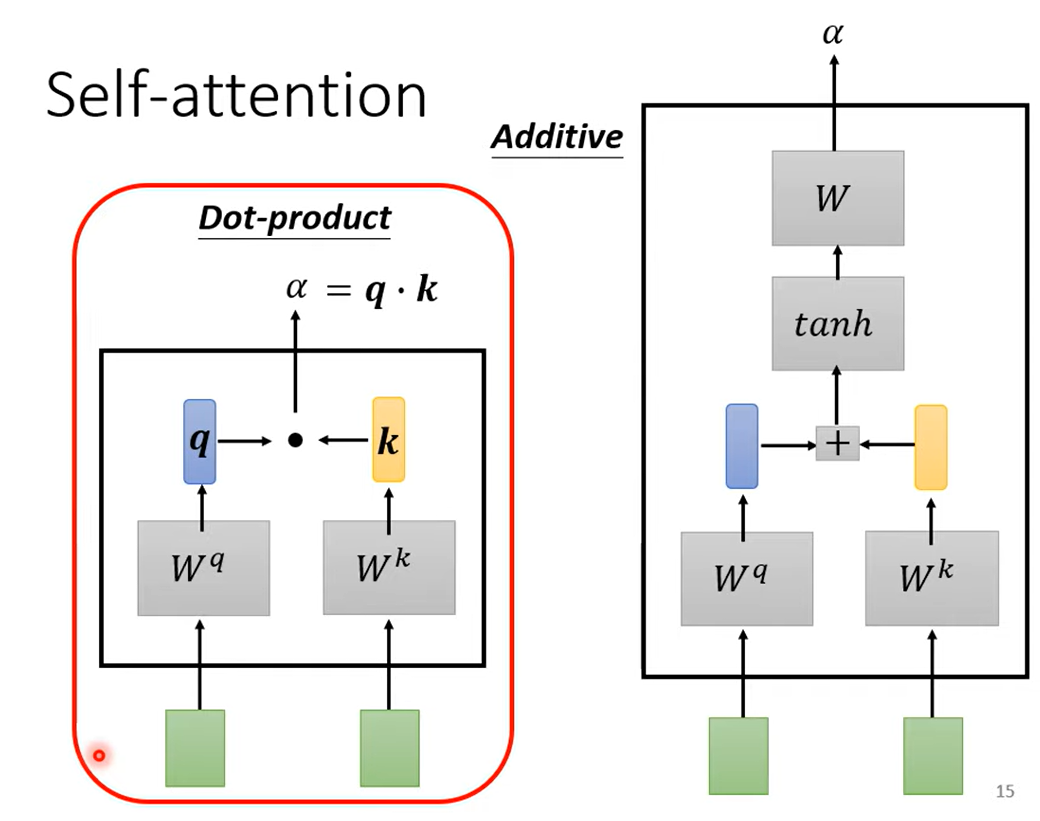
以为句子中的单词标注词性为例。因为输出与输入是一对一的，因而这里有一个直观的想法，即有一个FC，使得词汇输入模型后得到一个label。但是这个想法有明显问题，即词的词性是与上下文紧密联系的，因而我们要考虑不同词汇间的相互作用——这样就引出了想法，即设置一个window，可以覆盖住固定数量的词汇。随后将同一个window中的词汇进行一个交叉链接的操作，使得相互作用可以被学习。但这个思路也有问题，即当一个词的词性需要整个句子的作用才能显现的时候，我们需要窗口设置的非常大，才能使得不同词汇对于某词汇的影响被FC表达，但这也意味着模型复杂度的急剧上升以及更高的过拟合风险。

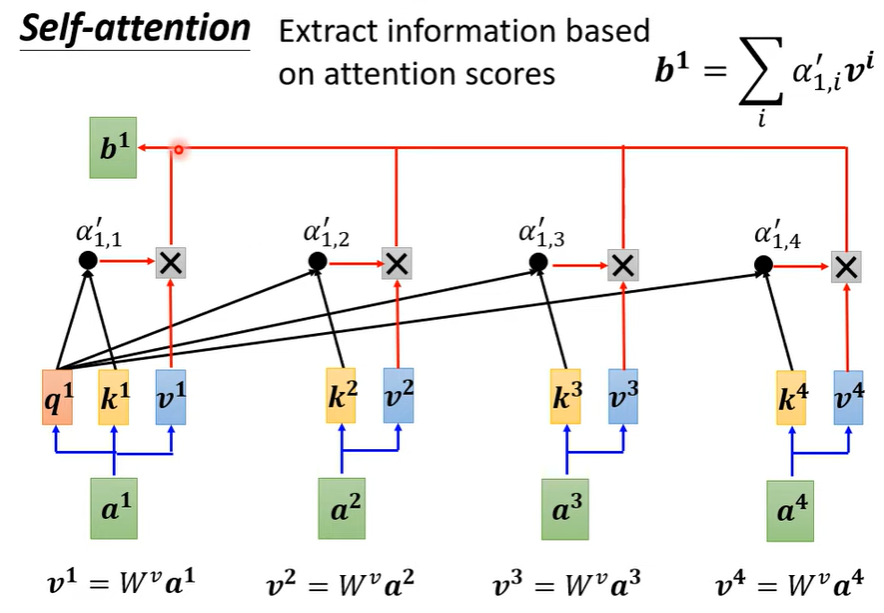


**Self-attention 自注意力：**

想法：类似一个预处理操作，这里是将每个vector传入到Self-attention之中，然后每个vector可以得到一个对应的新的vector，这个新的vector是对原vector考虑了全部输入vector后做调整的结果。Self-attention的结构类似一层神经网络。

Self-attention步骤如下：假设我们给定输入，现在我们想得到对应的输出，首先我们要做的是判断其他输入的vector与的相关程度（这个相关是针对于判定的词性的角度去阐述的），这个时候我们需要一个数值去度量两个向量间的相关性，因而不妨构造一个“内积”（如下图，最常用的是Dot-product模式，即将两个向量乘以一个特定矩阵后做哈达玛积），算出交叉的各个值。

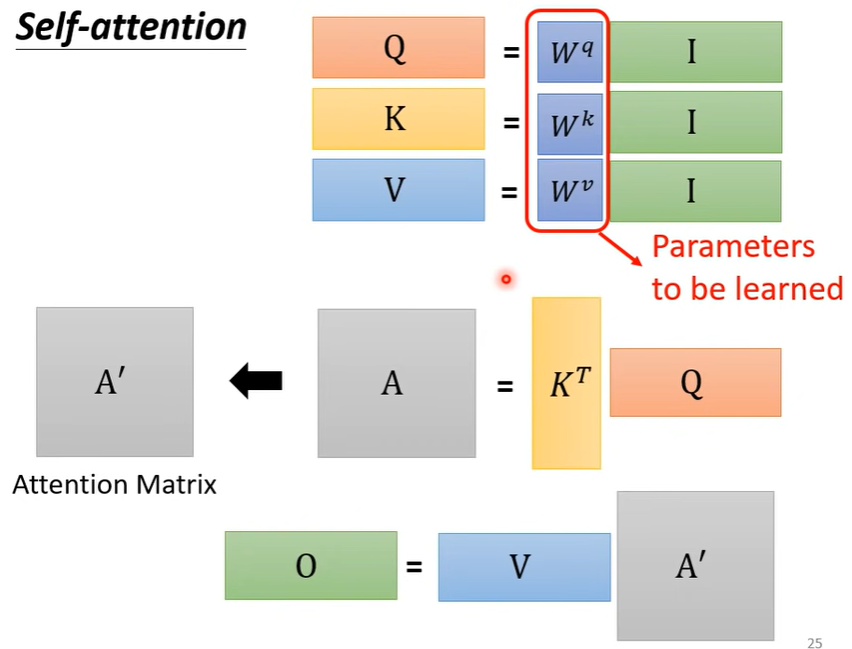




这里是将乘在左侧，乘在其他a左侧，q指的是query（询问），k指的是key。假定是与计算的attention score记为（当然，在实际工程中，也要计算自己与自己的关联性，这里指），随后对这几个attention score进行softmax操作（当然也可以是别的），以得到；随后对到左乘后再以为权做和，得到作为Self-attention层的输出。

**影片2 自注意力机制 (Self-attention) (下)**

***运算过程的矩阵化***



其中，，是输入的四个向量；指的是,对做逐列softmax后，得到为Attention Matrix；随后用矩阵左乘，即得到输出矩阵。由此，我们需要训练参数的就是三个W矩阵。

***Multi-head Self-attention 多头自注意力***

想法：