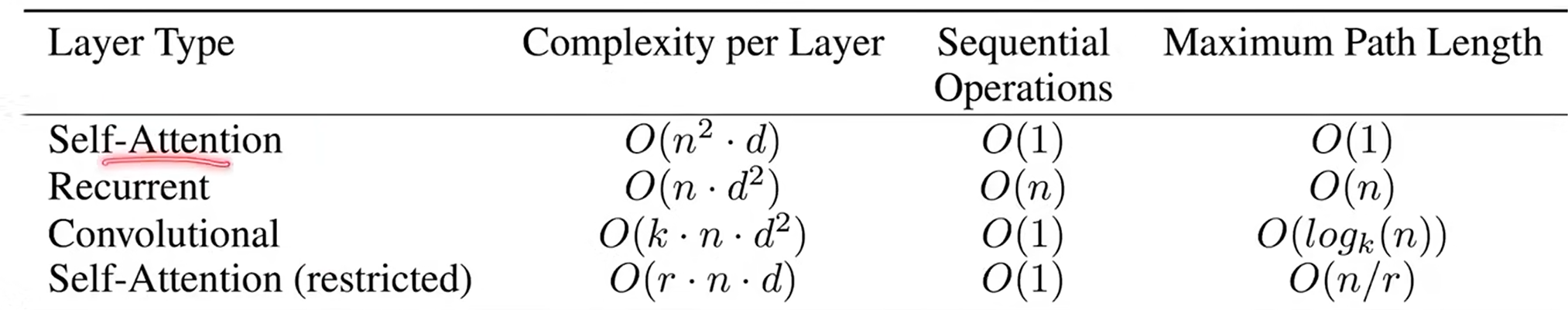
RNN（循环神经网络）：（早期用于机器翻译（如LSTM）、文本生成）时序地运行，并行度低计算性差，如果序列长那么需要大的内存开销来存储h（t），早期时间步的信息可能因多次循环而丢失（LSTM/GRU缓解但未彻底解决）。

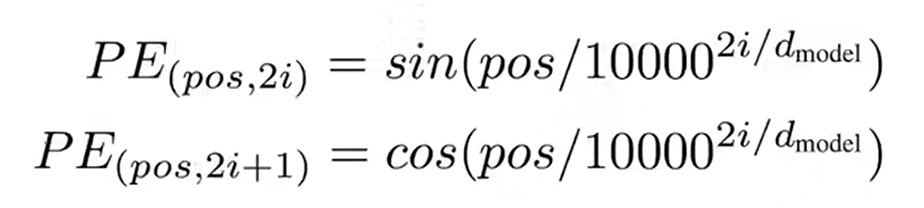
CNN（卷积神经网络）：（文本分类（如TextCNN）、图像识别）长距离依赖需要神网络，浅层CNN只能捕捉局部特征，需堆叠多层扩大感受野（可能引入冗余计算）f

Transformer：可以并行运算，只依赖于注意力机制来实现从encoder到decoder的架构，用多头注意力机制输出不同模式的结果

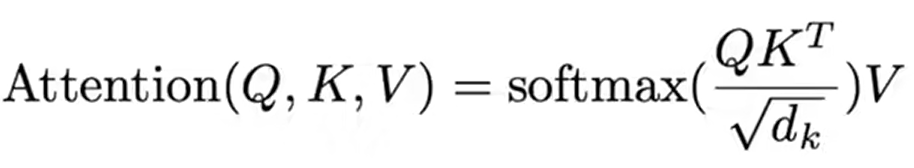


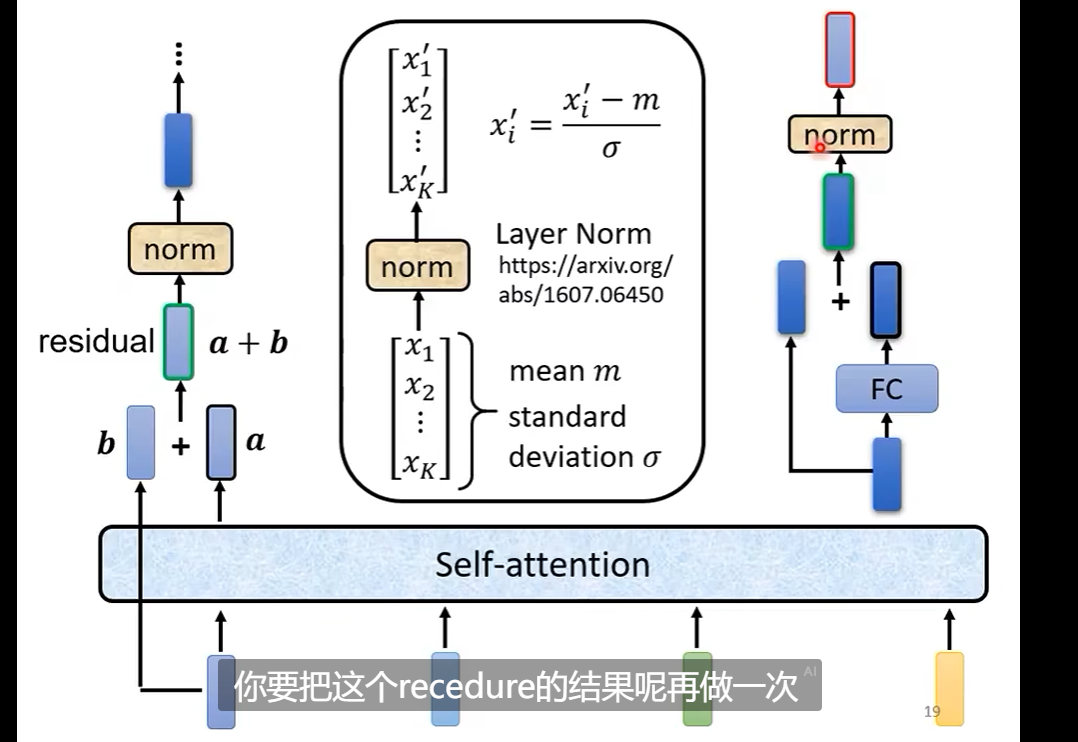
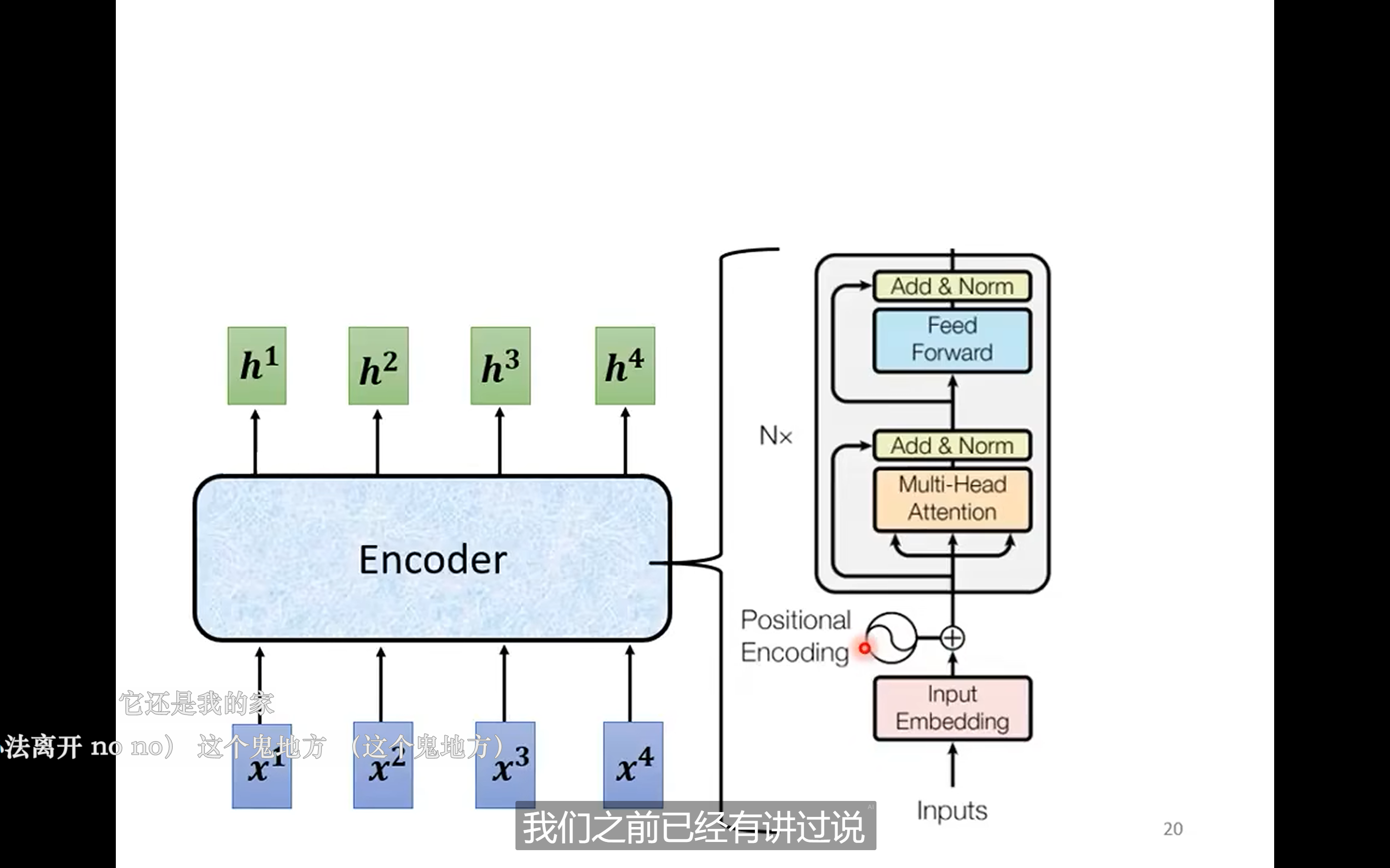
Seq2seq（输出序列长度可以与输入序列长度一样、输出序列长度为1、输出序列长度未知由机器自己决定）

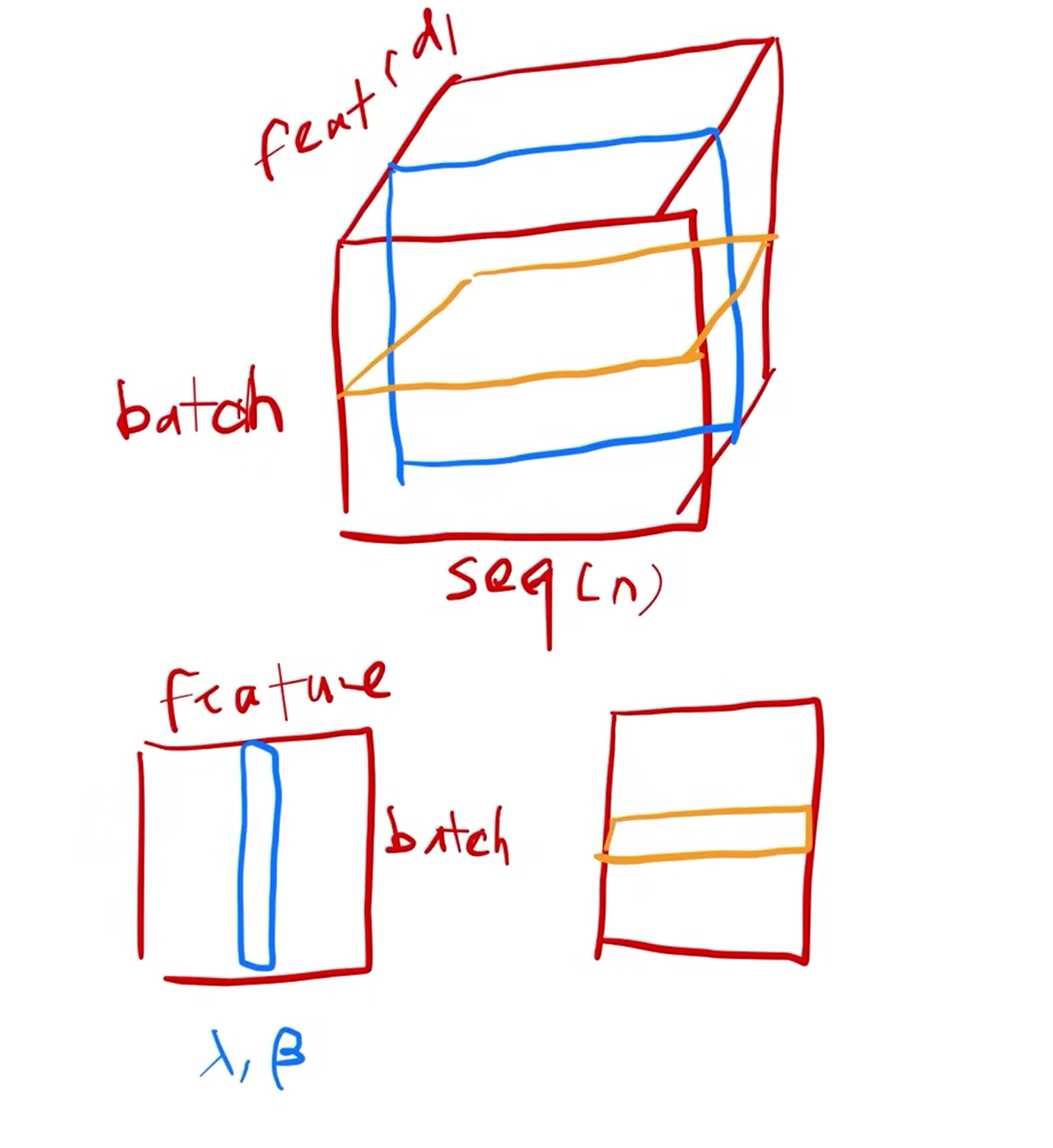
Positional encoding层：记录时序信息，与嵌入层得到的向量相加



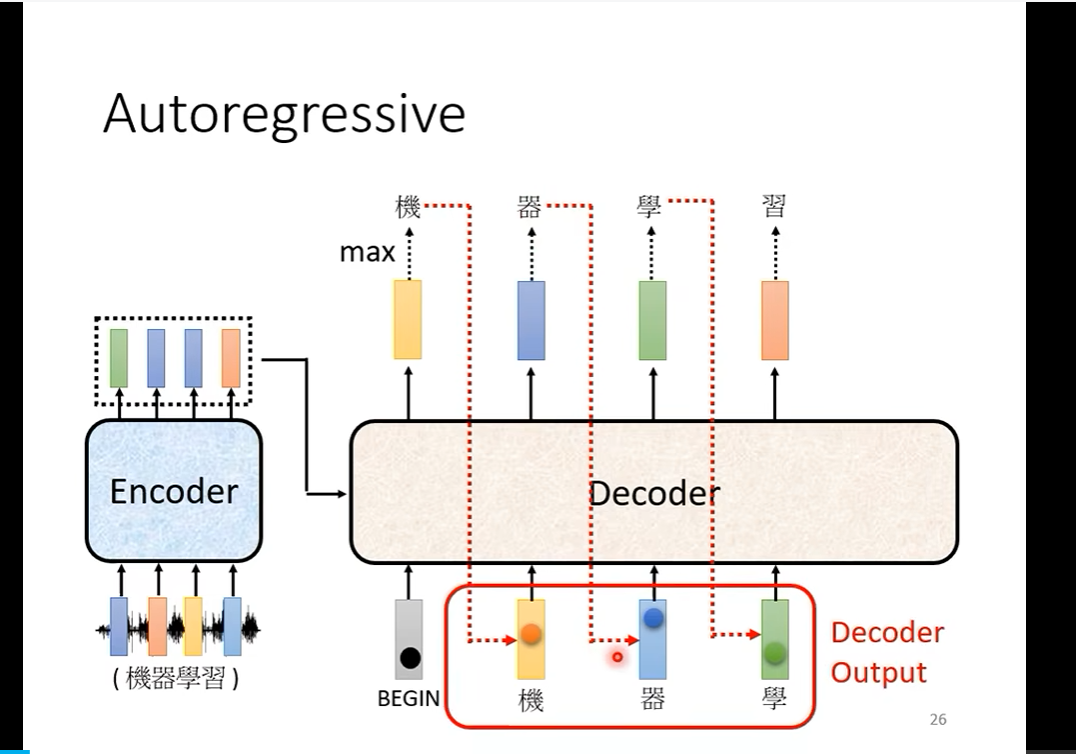
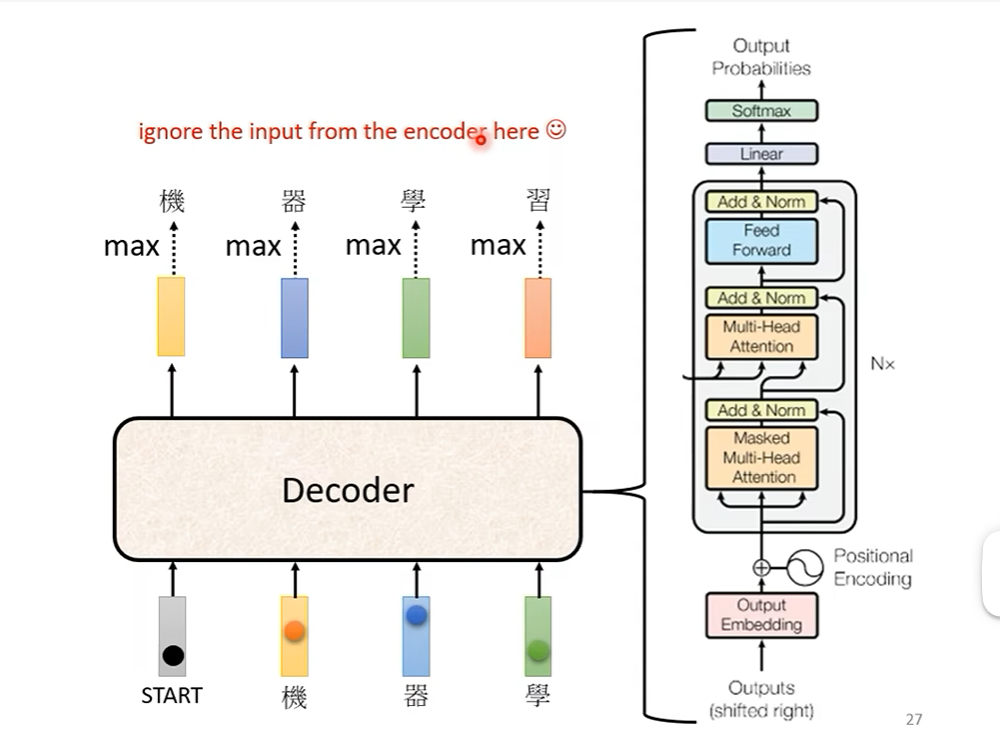
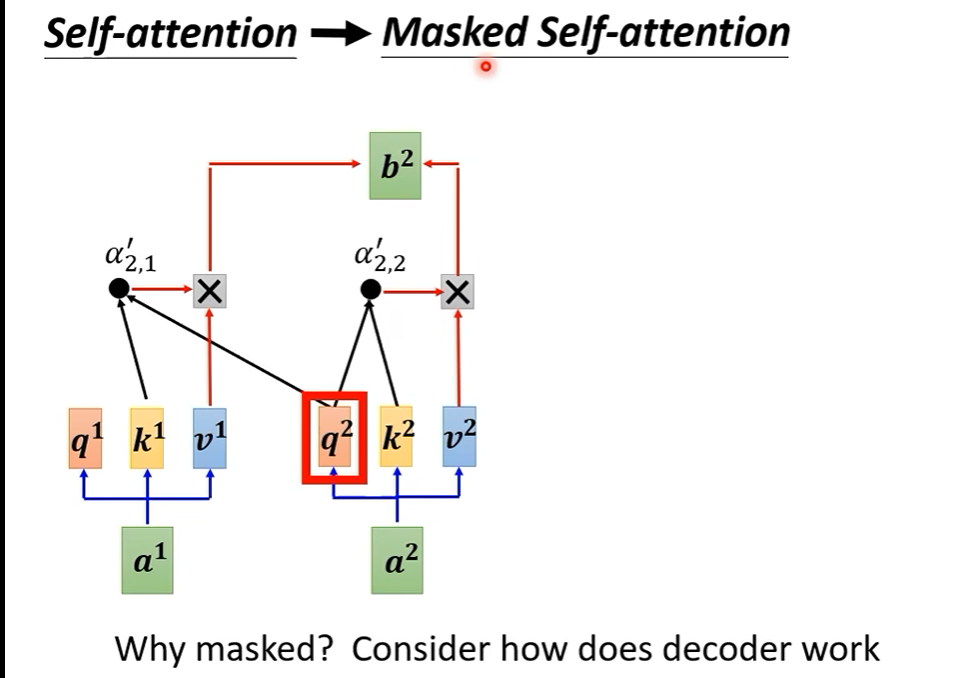
以下是transformer的encoder设计：





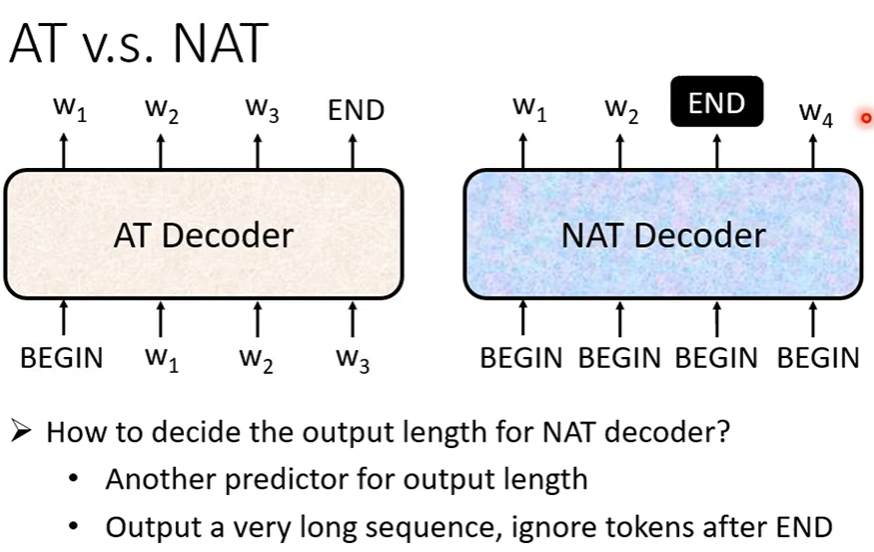
Batch norm和layer norm的区别：（蓝色为batch，黄色为layer，下面是二维，上面是三维）

Decoder：

此处的masked意味着输出的b2只能考虑a1和a2，因为a1、a2、a3、a4是一个个顺序输入的，要输出b2时能参考的只有a1、a2

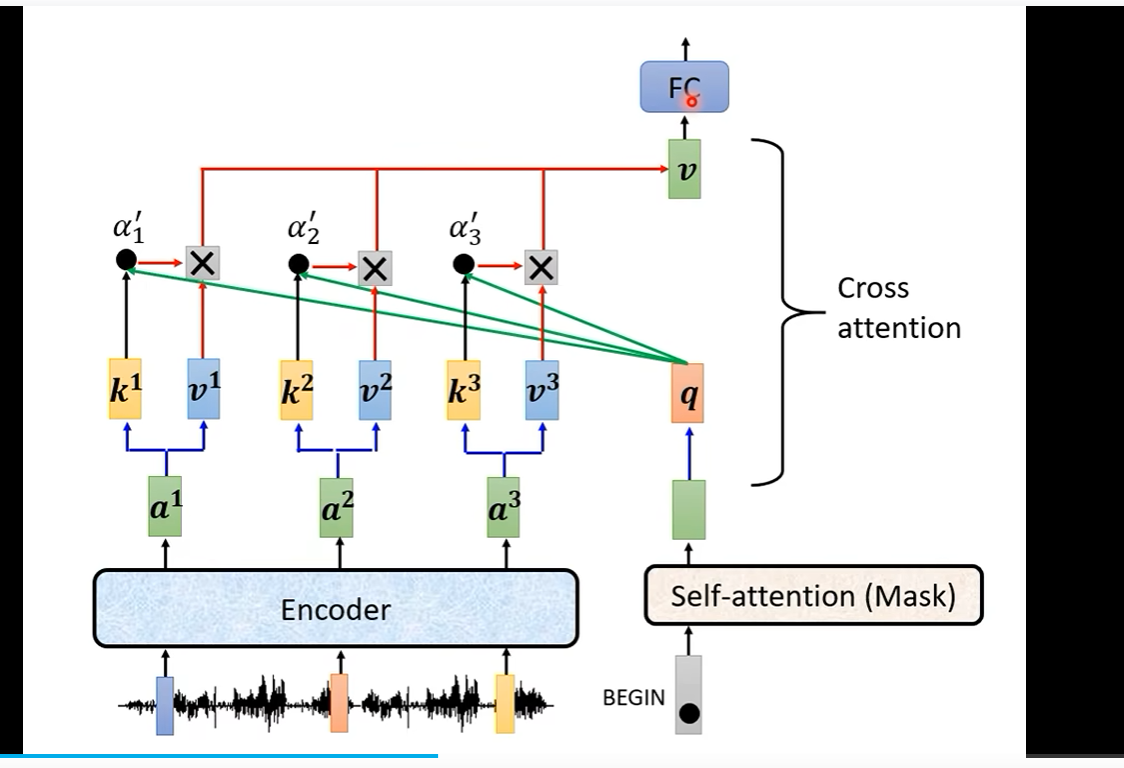
看到begin开始输出，为了让输出可以在合适的时候停下来，需要有end，其实begin和end用的是同一个符号



NAT decoder相对于AT decoder的优点是：平行化（于是速度快）

NAT可以控制输出序列长度

Cross-attention的运作过程：通过decorder里的begin与encoder输出的向量做注意力分数运算，然后将这个结果作为fully-connected network的输入



如何训练模型：

Transformer实现的是一个分类的功能

训练的时候要输入正确的数据来训练，即输入“机器学习”作为decorder的输入，训练他输出“机器学习”，但有时也需要scheduled sampling输入一些错误数据，以免面对真实的输入时措不及防

训练的tips：

有时机器不需要创造，而是从输入中复制一些内容作为输出，譬如人机对话、机器对文献做摘要

语音辨识、合成很适合用guided attention

Beam search：二元中输出分数较高的哪一个，这个技术能找到一条总体上较好的路径

语言合成时训练decoder也许加入噪声效果会更好

用BLEU scores来评估模型，用cross entropy来训练