

Transformer

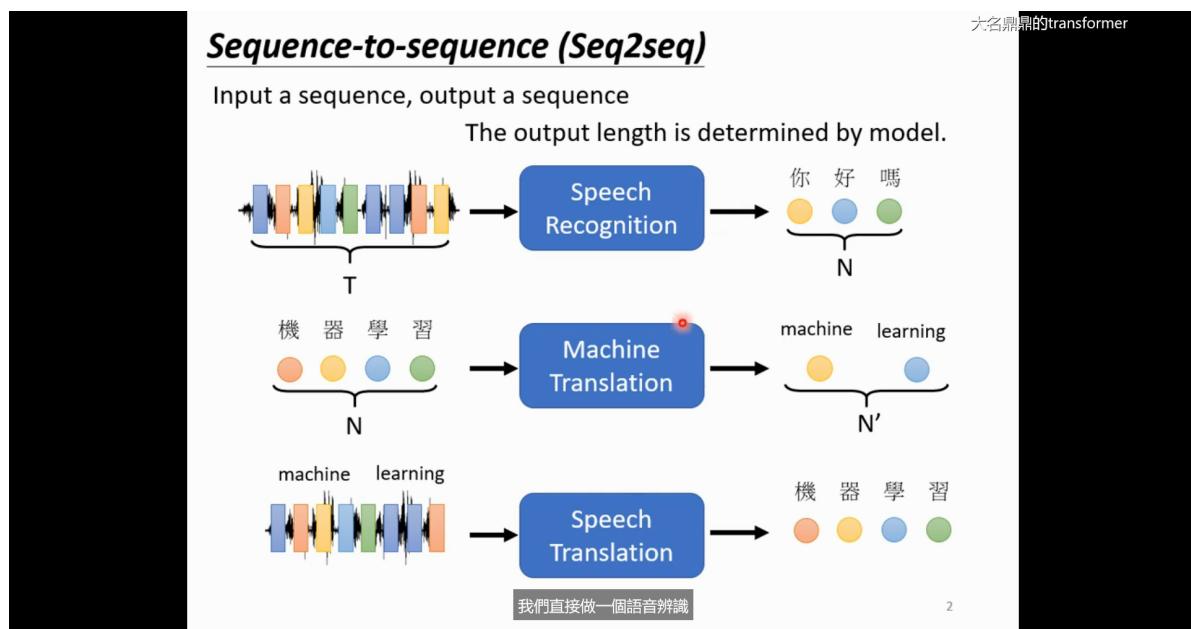
Transformer

Encoder用的Self-attention

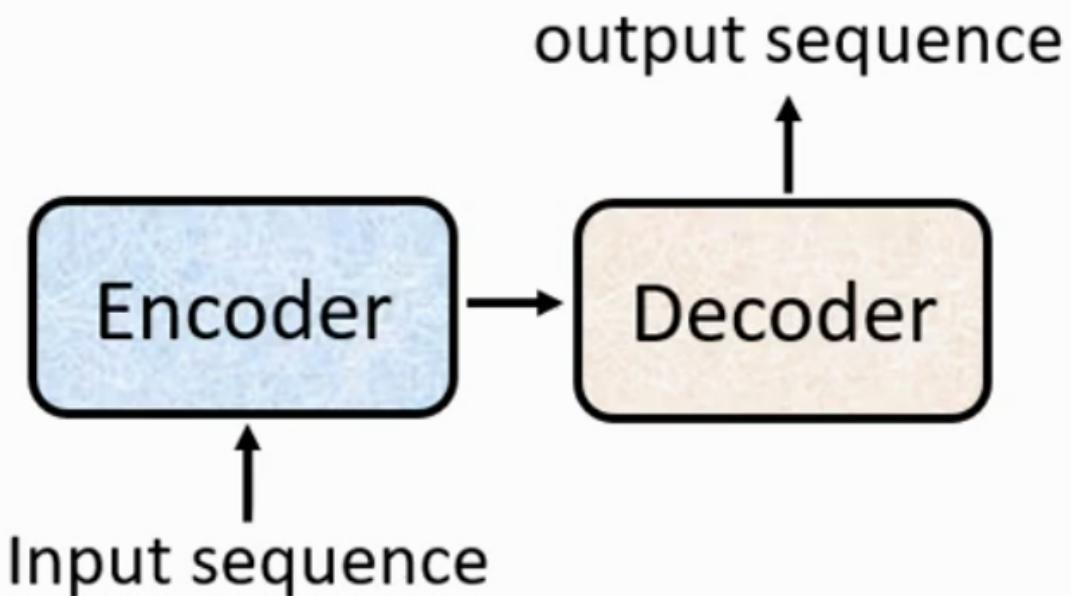
Seq2Seq

Seq2Seq的model

输入一个单词序列,输出一个单词序列,输出的长度取决于模型

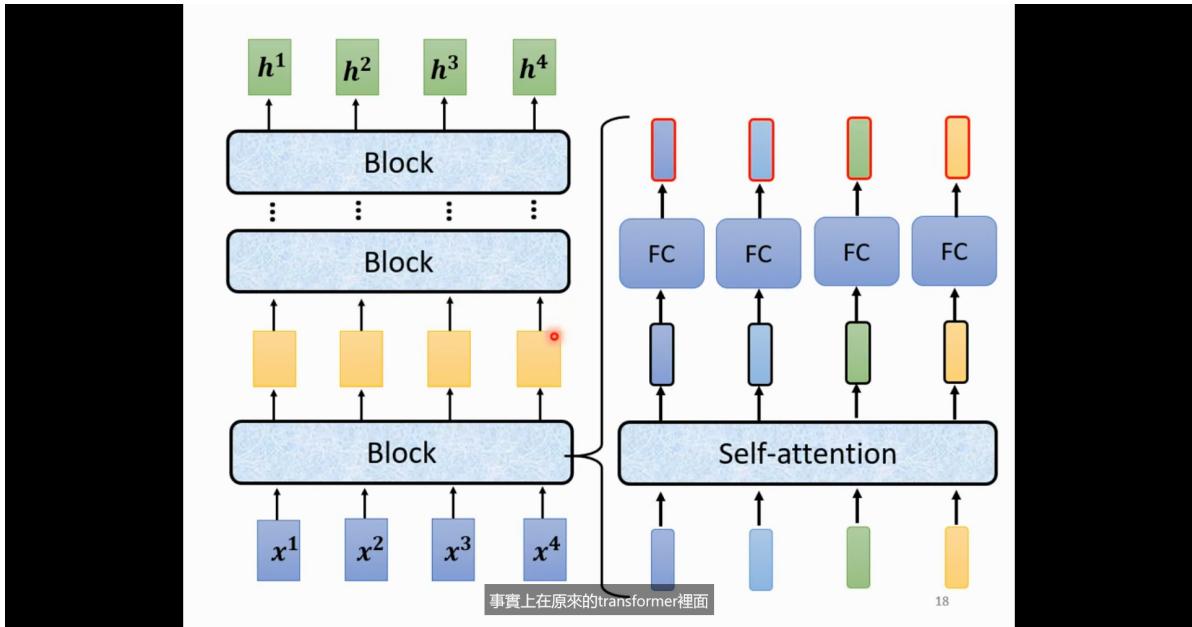


Seq2seq

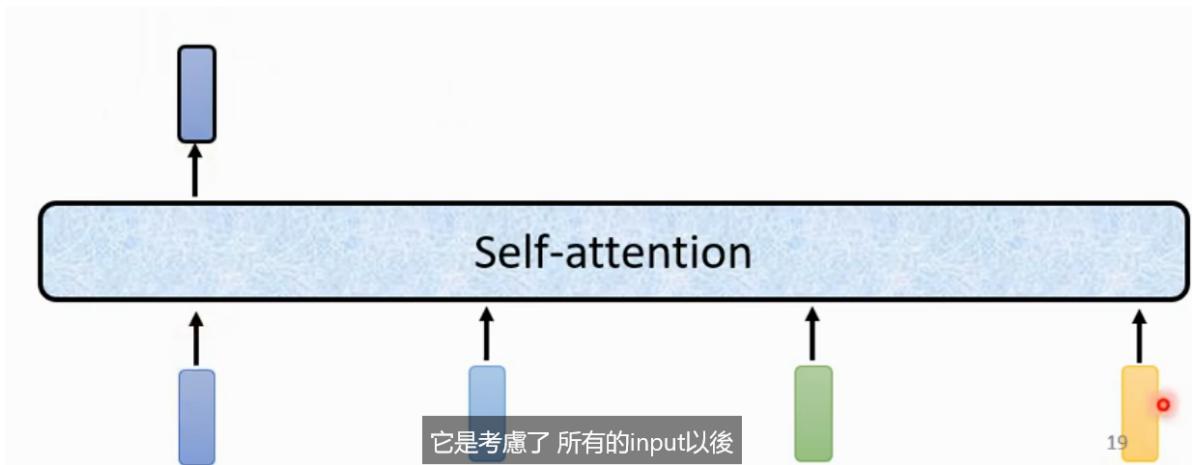


Encoder

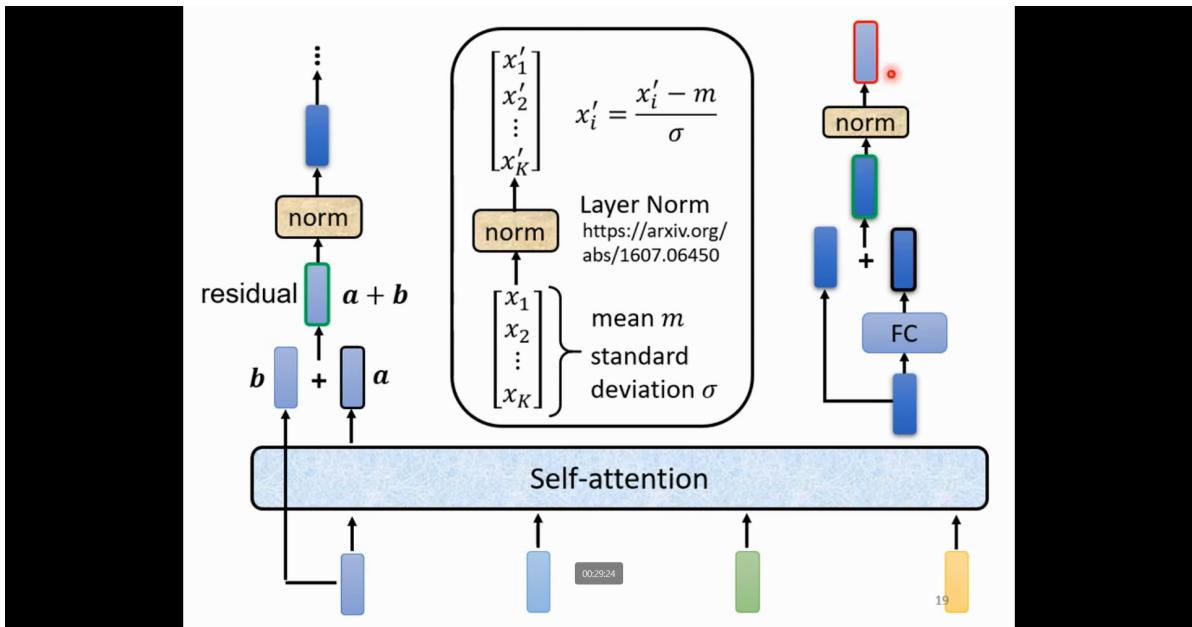
给一排向量,给出另一排向量(相同长度)



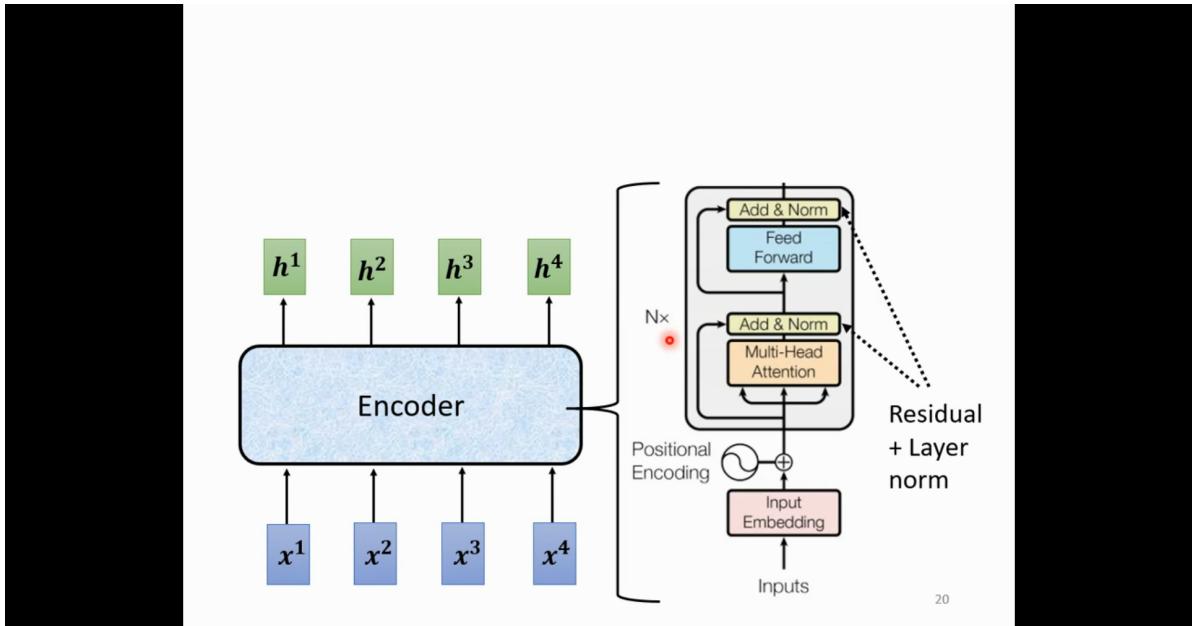
每一个经过Self-Attention的输出vector都是考虑了所有的输入之后得到的结果



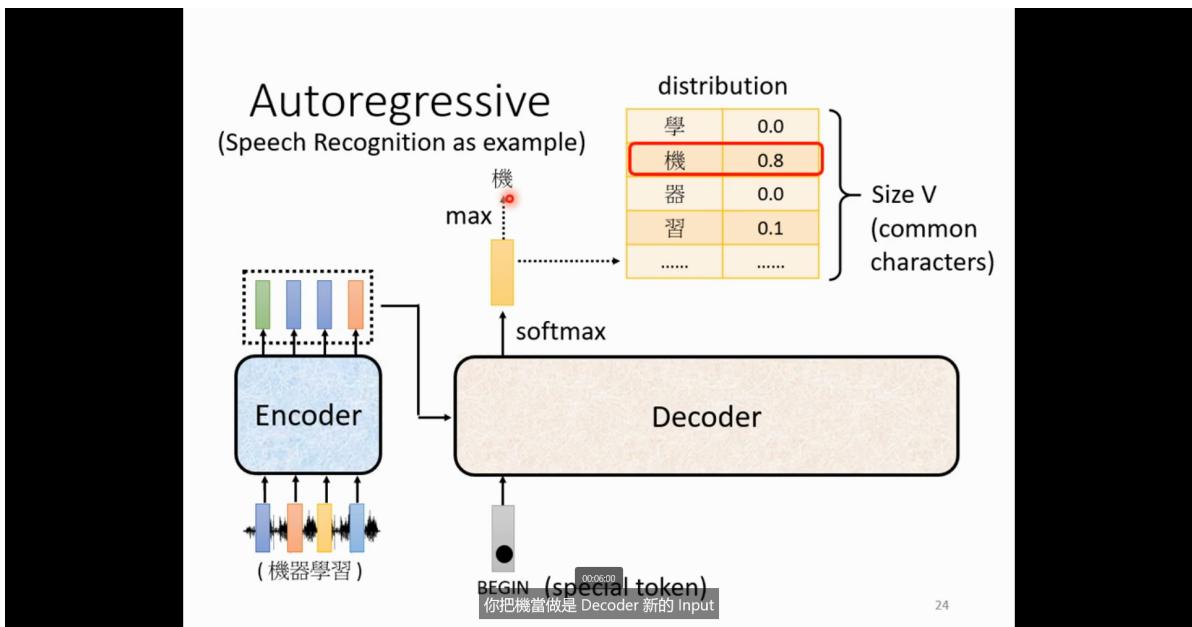
Transformer的Encoder如下图所示, 经过Self-Attention的输出是综合了所有输入的, 然后这个输出的结果再加上原本的输入(residual)再经过Layer norm, 经过FC层, 再加上原本的(residual)经过norm后再输出



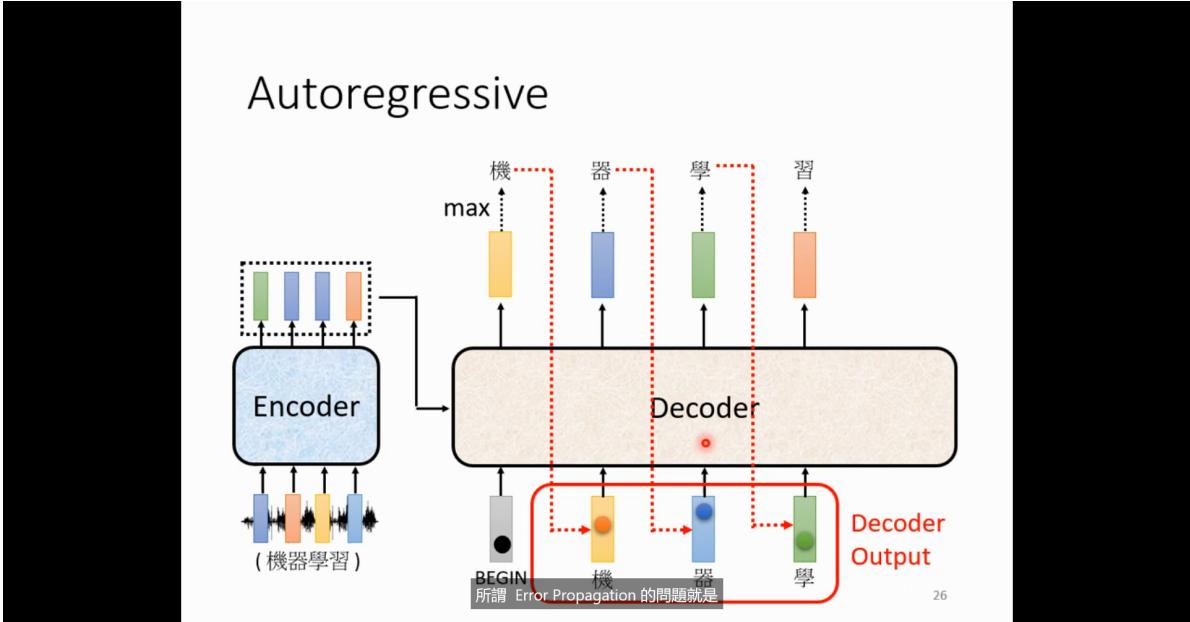
论文中的Encoder模型(Bert就是这个Encoder)



Decoder(Autoregressive)

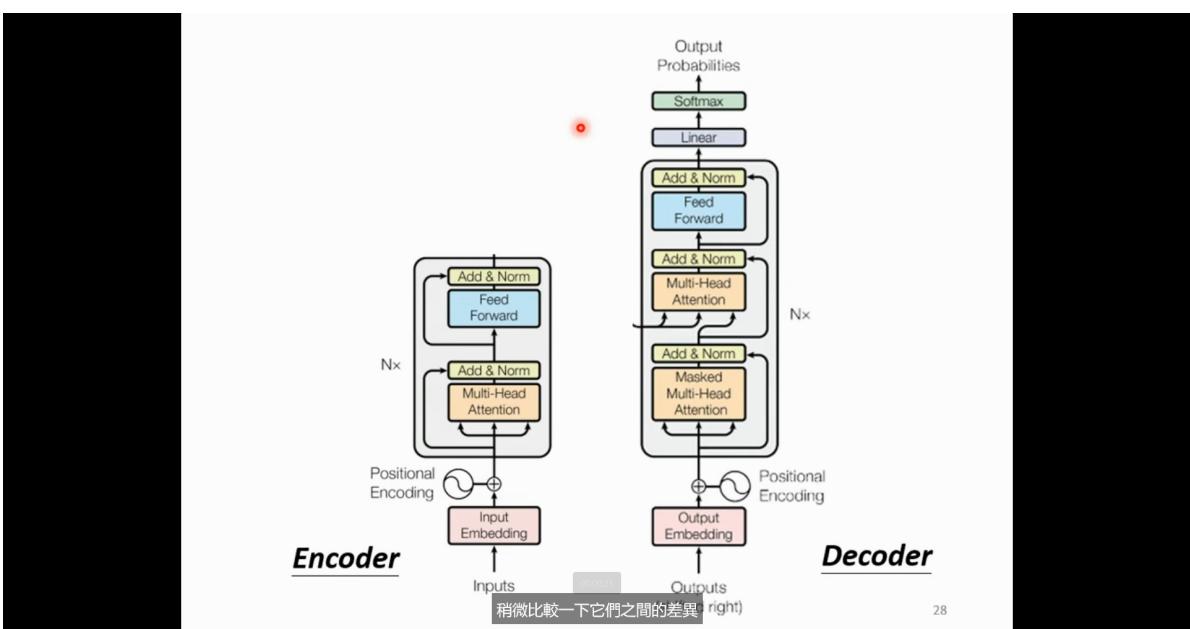
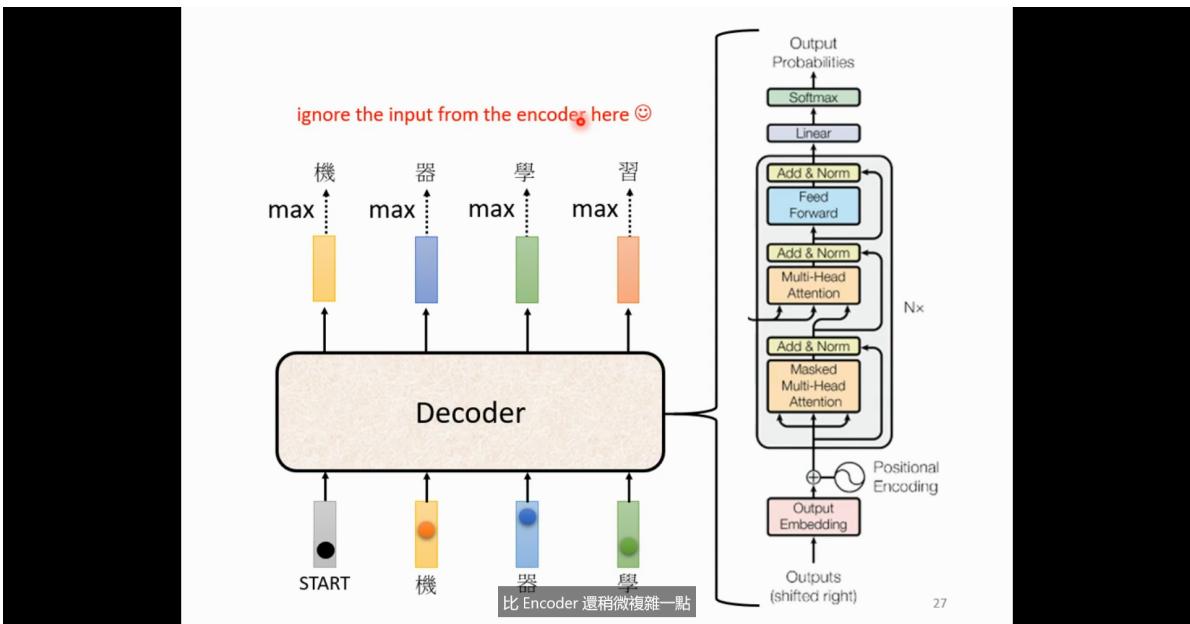


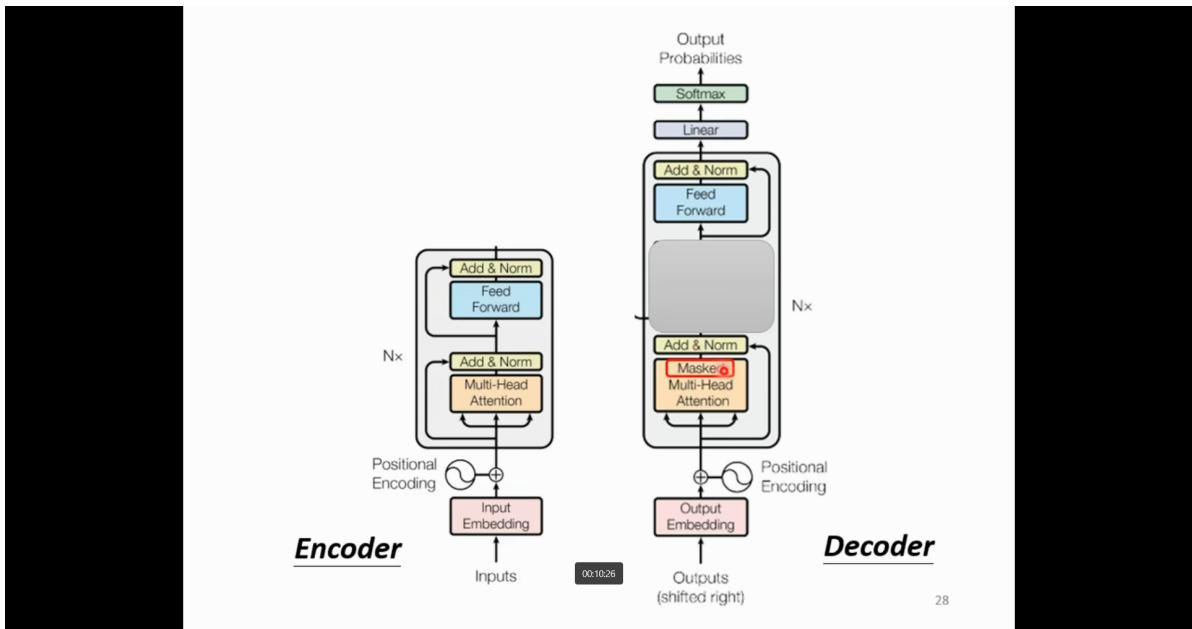
Autoregressive



Decoder看到的输入其实是上一个时间点的输出

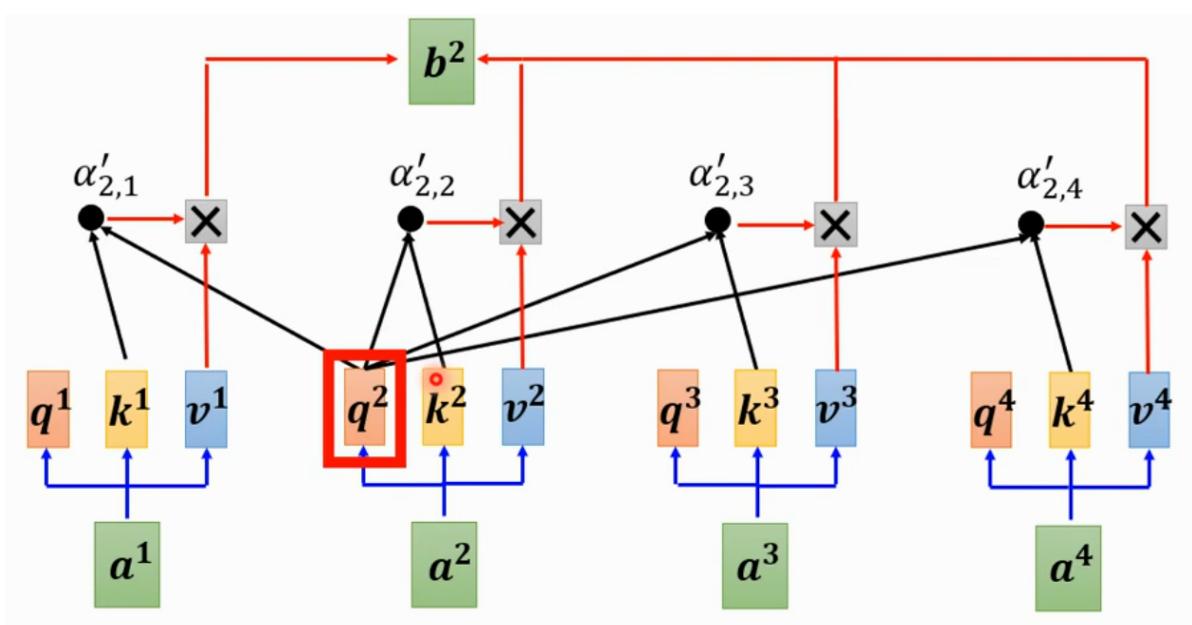
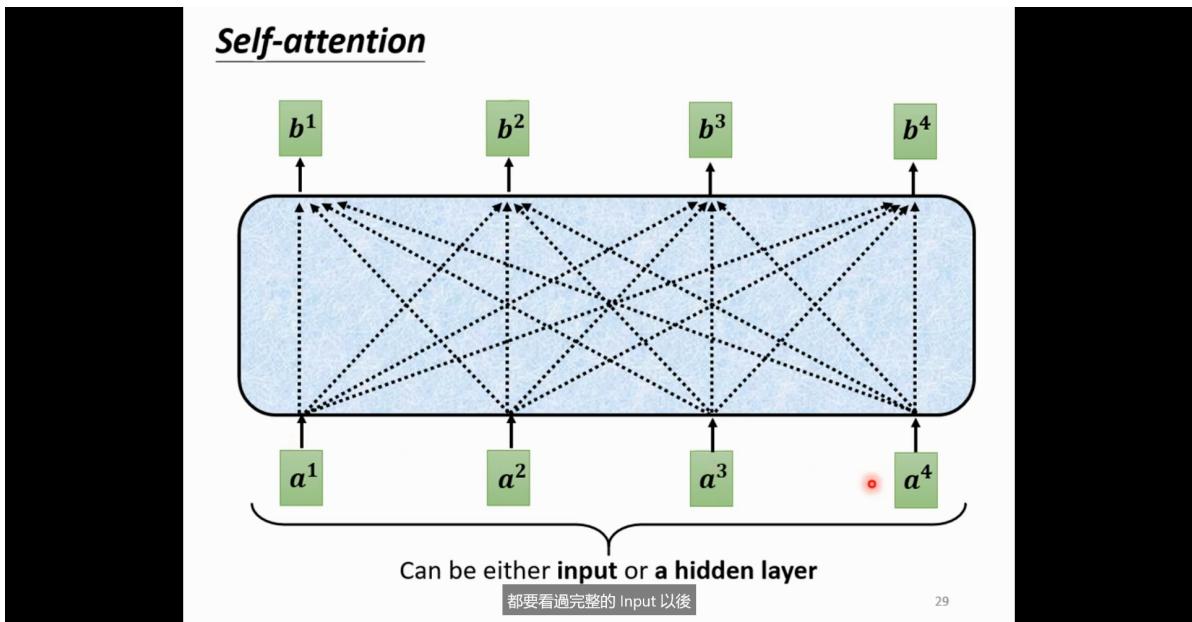
Transformer里的Decoder结构





Masked

原始的Self-attention,每个输出都取决于每个输入

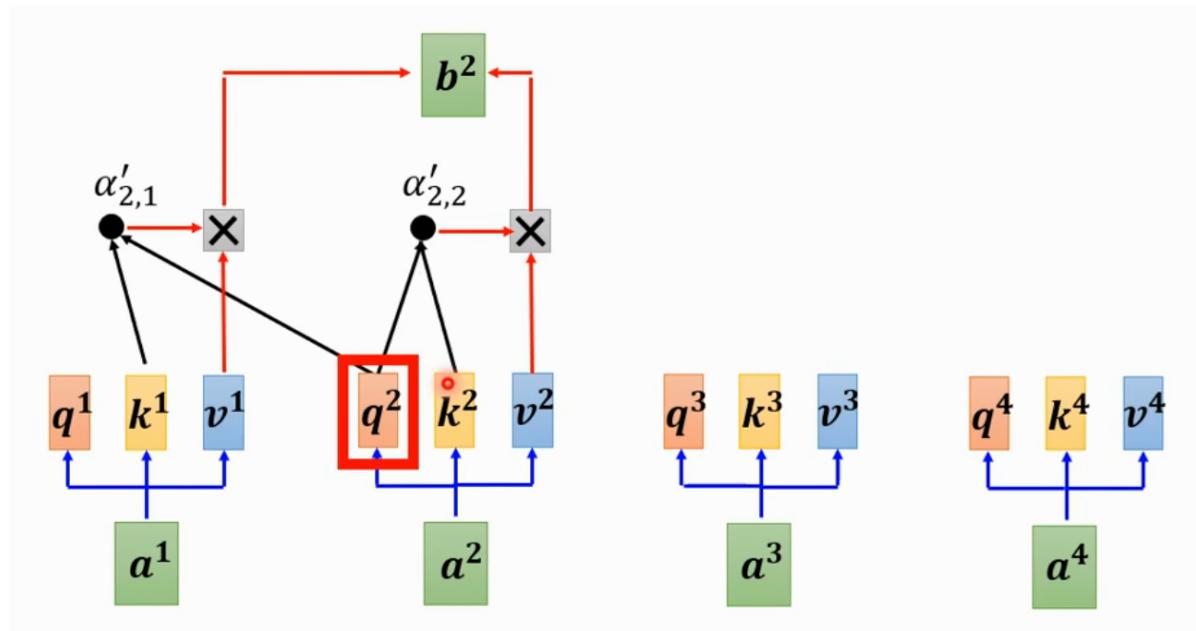
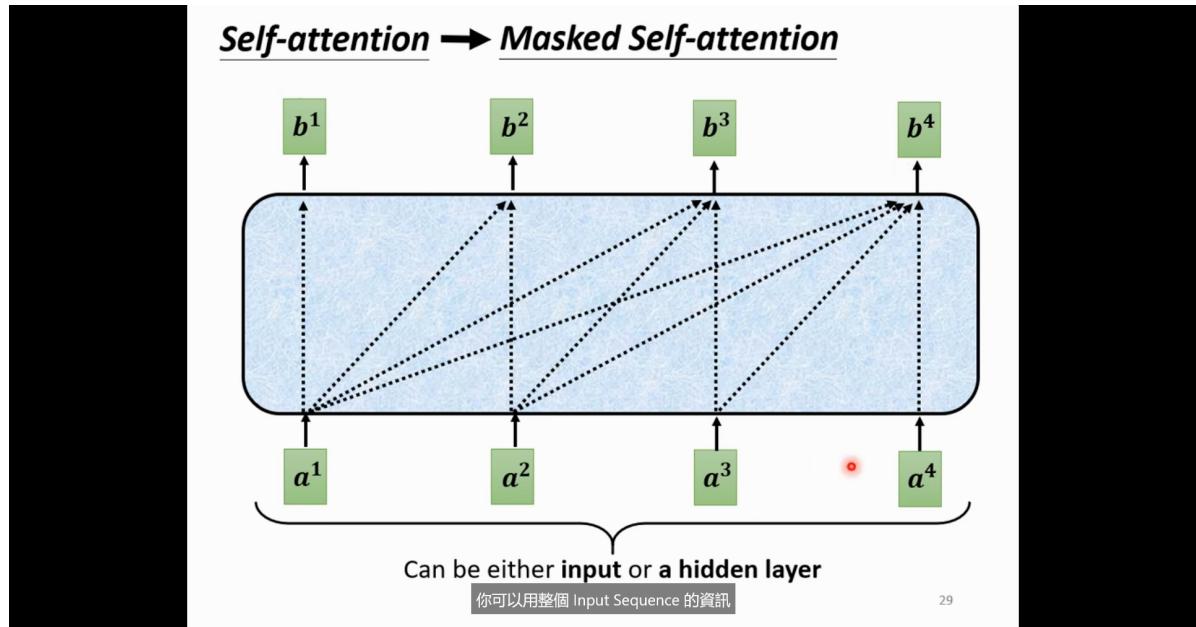


Masked-Self-attention

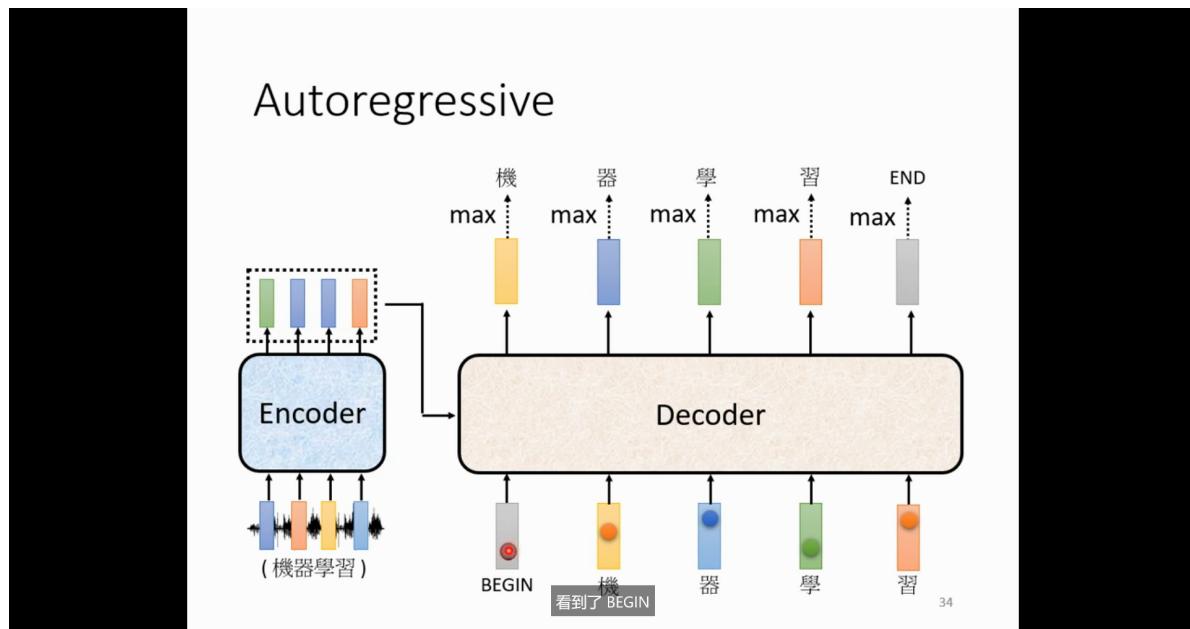
每个输出只能考虑这个时间段之前的输入

因为对Decoder而言,他每次输入一个不是一排,只能考虑已经发生的事

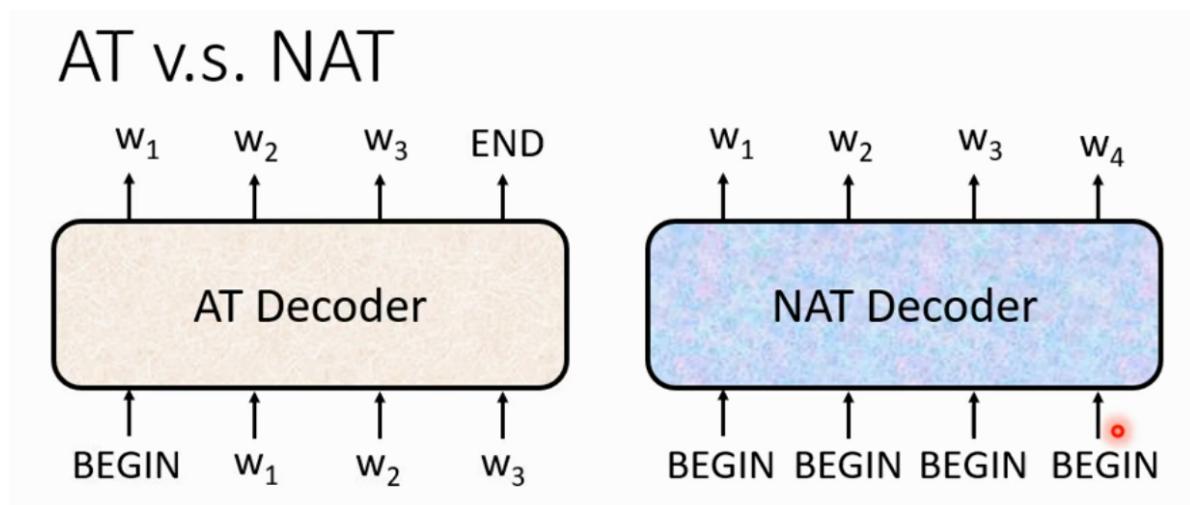
计算 b_2 时只有 a_1 和 a_2 ,没有办法考虑 a_3 a_4



词袋长度,所有的字加BOS,END



Decoder(Non-Autoregressive) NAT



AT:每次输入一个,输出一个

NAT:每次属于一排,输出一排

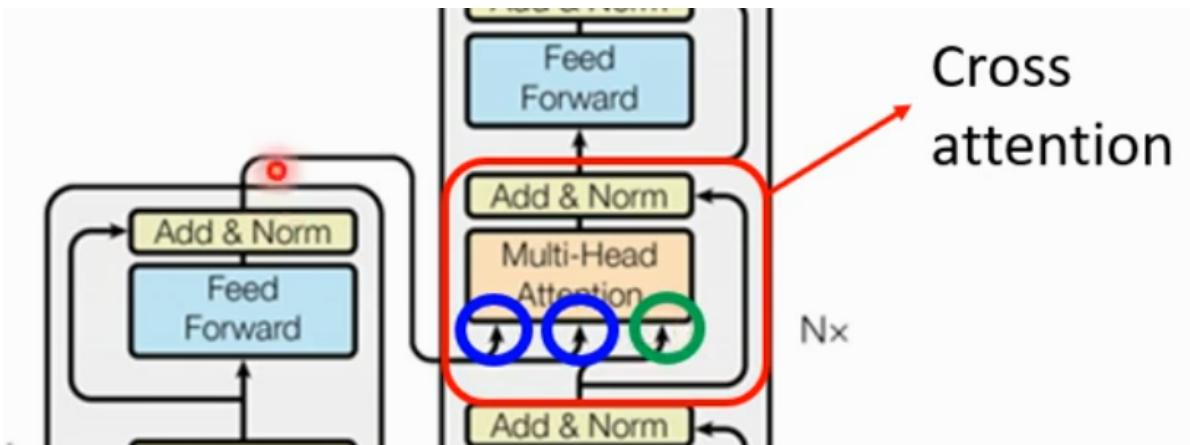
Encoder-Decoder中间传递消息

Cross Attention

两个信息来自Encoder一个来自Decoder

K,V来自Encoder

Q来自Decoder



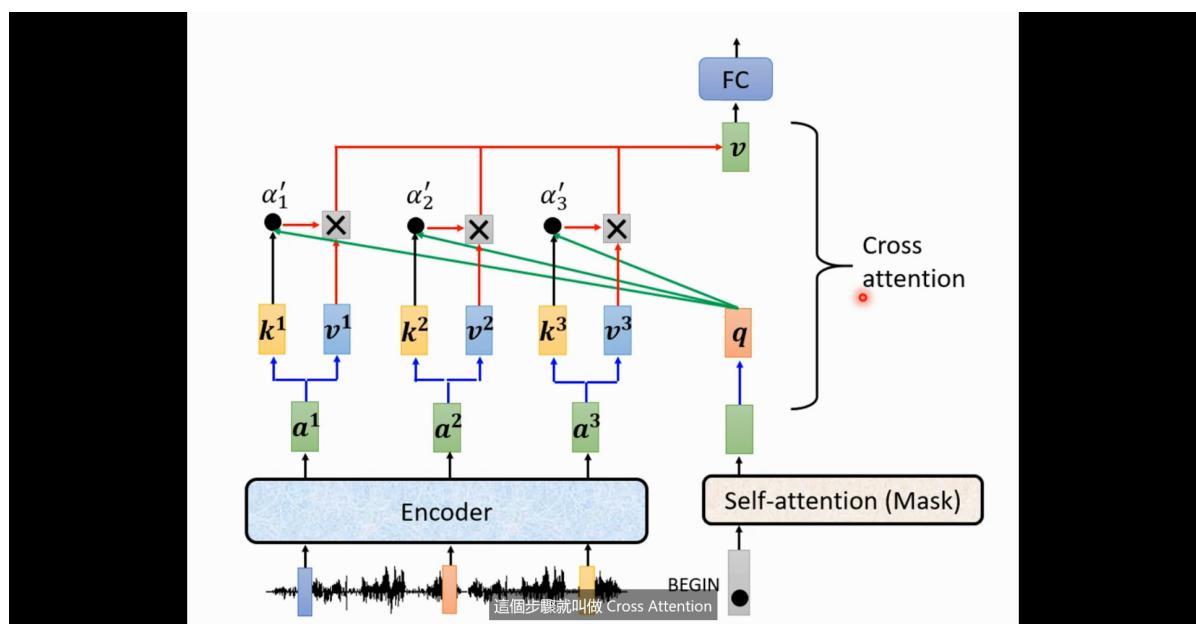
一个输入经过Encoder后得到的输出乘上一个矩阵做Transform产生k,v

另一边产生q

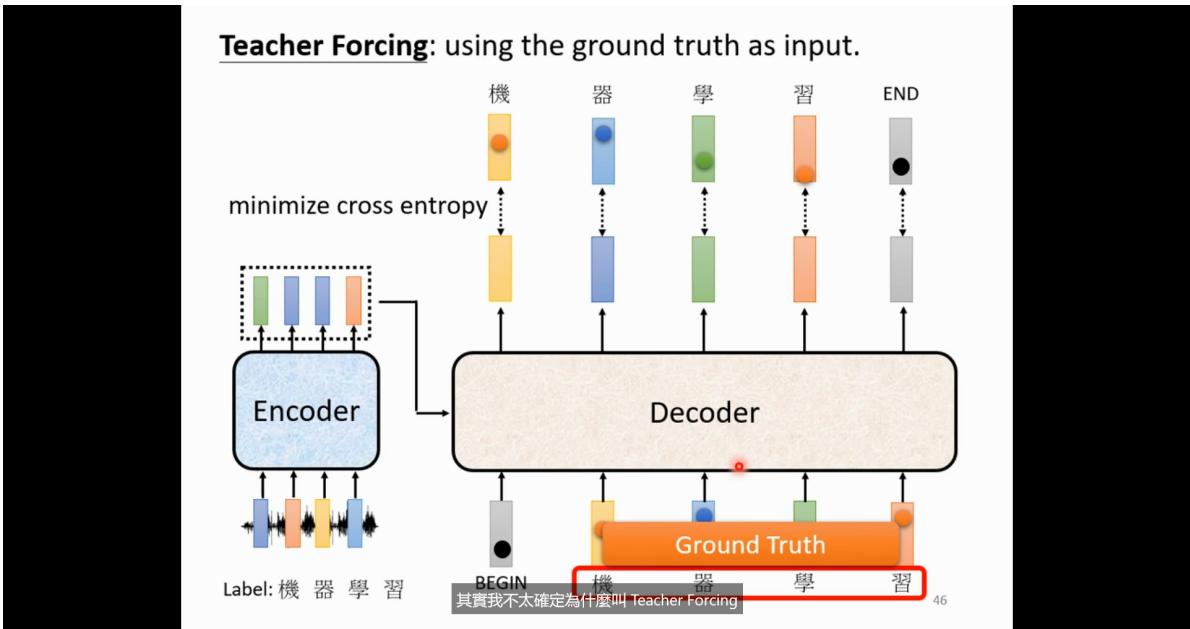
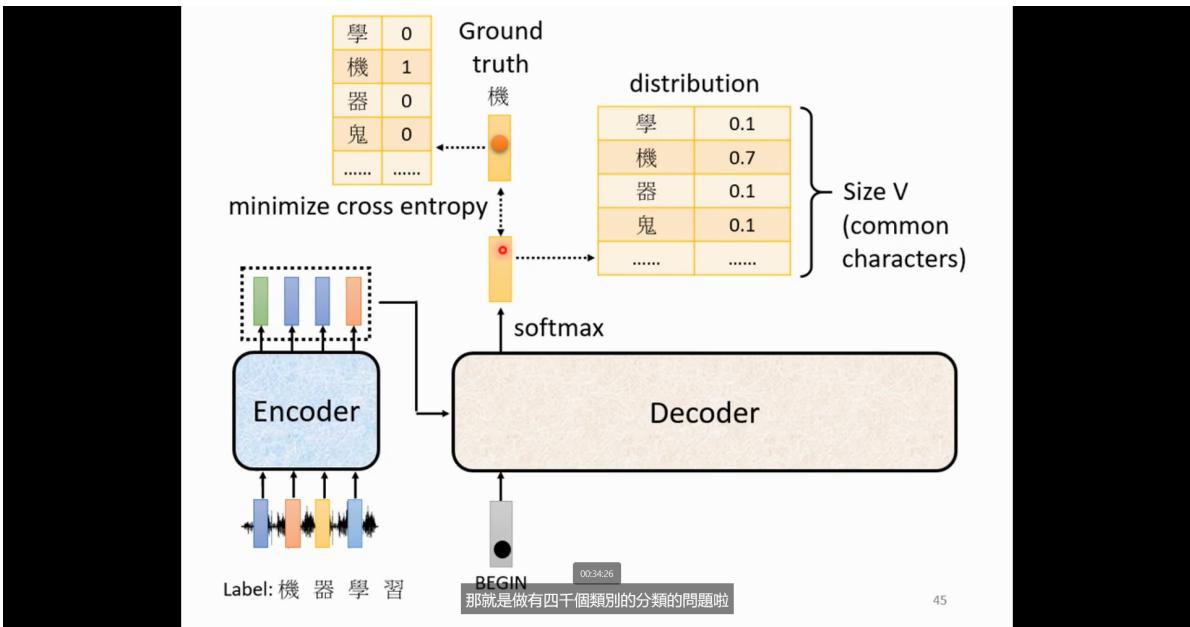
q和k计算Attention的分数得到 $a^{'}$

$a^{'}$ 再和v相乘,加起来得到v

v经过FC层



训练



总的交叉熵损失最小

Tips

复制一些文字

聊天机器人,对话机器人,摘要

Guided Attention

语音合成,语音辨别

attention是有顺序的

Beam Search

局部最优,不是 全局最优

Scheduled Sampling

可能会出先学的都是正确的,但出错之后就一直错