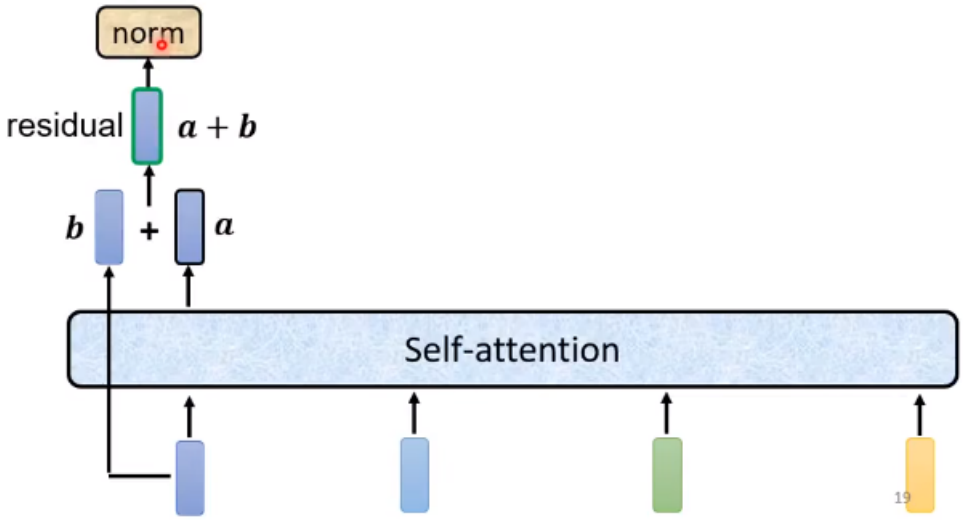
Transformer

1 Transformer中的一个模块

Transformer中有1种全连接层和3种自注意力层，每一层都有如下结构：

1.1 残差连接

每一个注意力层都会进行残差连接：



实际就是将注意力层输出的结果与原始词向量相加，这就是残差连接。

- 为什么要进行残差连接?

  在一定程度上，网络越深表达能力越强，性能越好。但是，随着网络深度的增加，带来了许多问题，如梯度消散，梯度爆炸等。

  假如有如下函数，使其误差链式反向传播：

  $f' = f(x, w_f)$

  $g' = g(f')$

  $y' = k(g')$

  $cost = criterion(y, y')$

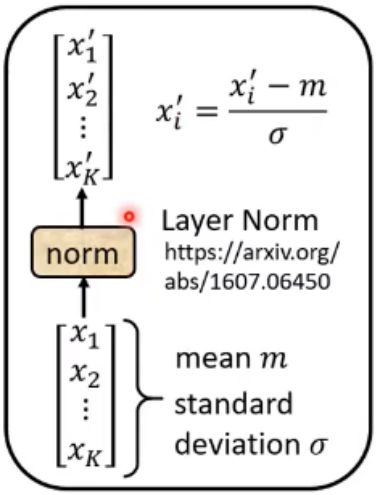
  链式求导结果为：$\frac{d(f')}{d(w_f)} * \frac{d(g')}{d(f')} * \frac{d(y')}{d(g')} * \frac{d(cost)}{d(y')}$

  一旦其中某一个导数很小，多次连乘后梯度可能越来越小，**因此很可能出现梯度消失**，对于深层网络，传到浅层几乎就没了。但是如果使用了残差，**每一个导数就加上了一个恒等项1，**$\frac{d(h)}{dx} = \frac{d(f+x)}{dx} = 1 + \frac{df}{dx}$。此时就算原来的导数$\frac{df}{dx}$很小，这时候误差仍然能够有效的反向传播。

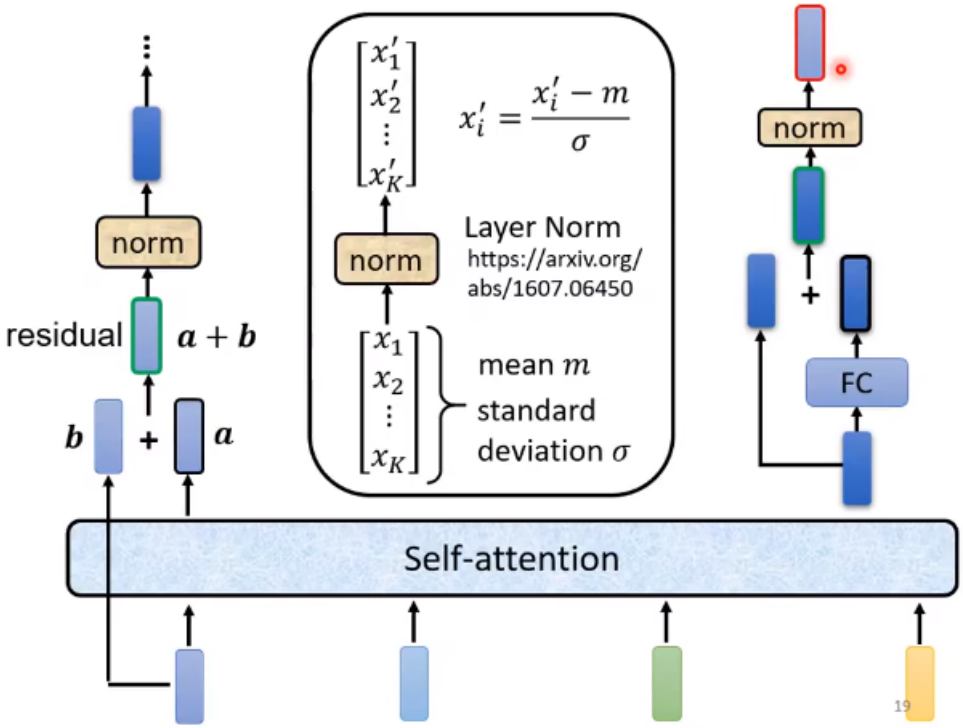
- 具体残差连接是如何使得神经网络变得很深的，可以阅读如下文献：

1.2 Layer Norm

对每一个输出向量标准化，而不是对一个batch标准化



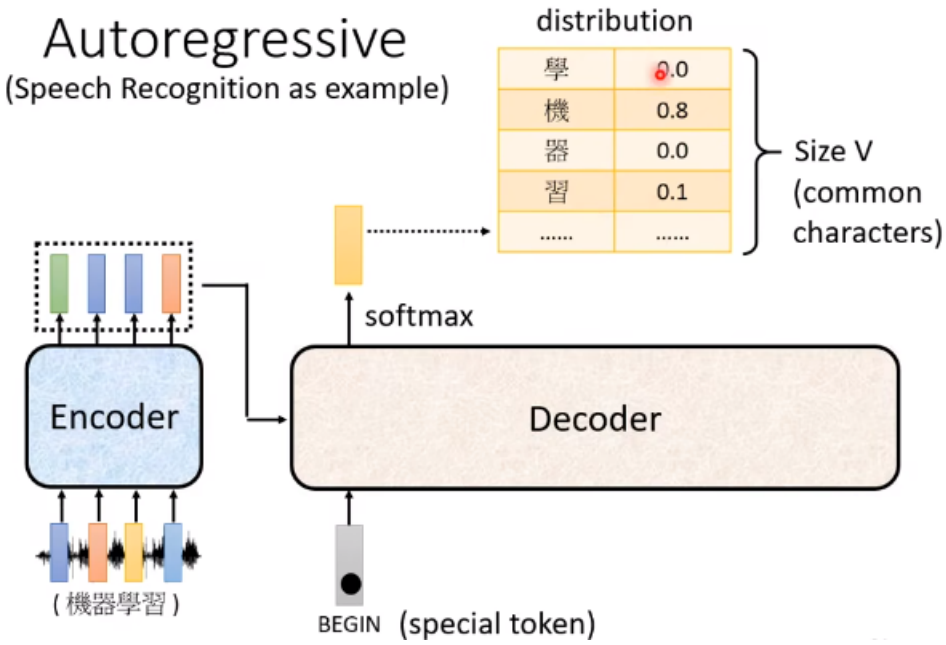
1.3 模块架构图



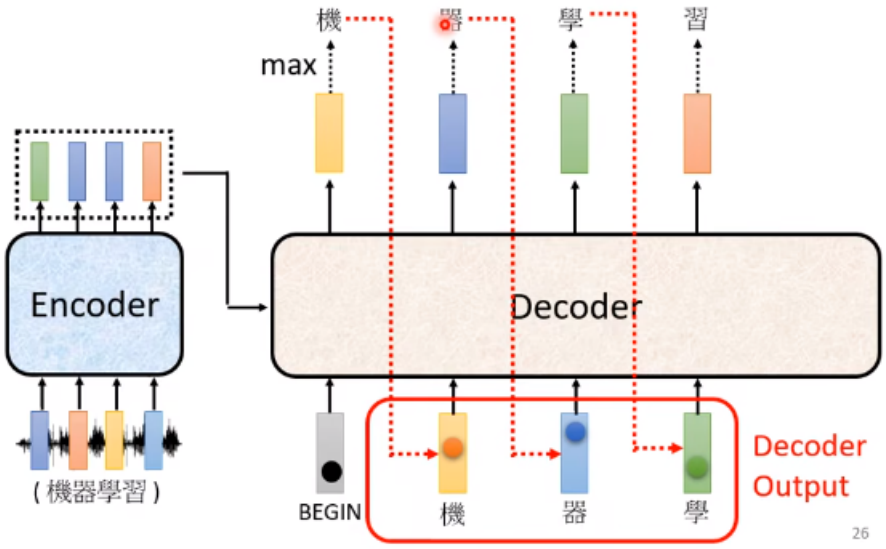
2 Decoder

2.1 Decoder大致流程

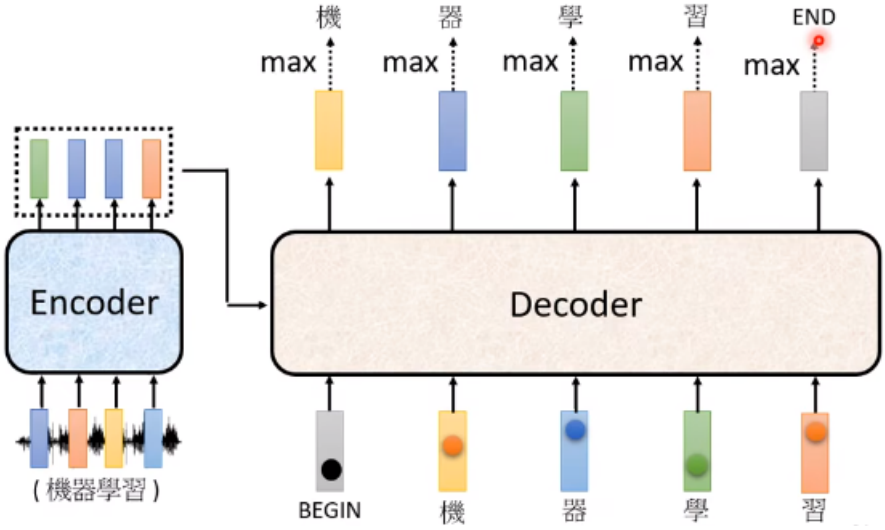
将Encoder的结果输入到Decoder中，循环输出得到每一个词的概率：



从一个Begin开始，每一个输出词都是下一个输出词的输入：

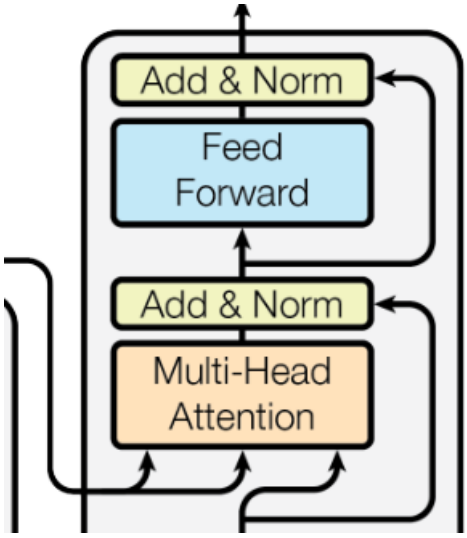


进行完特定的任务之后，Decoder会输出一个END字符，表示本次翻译停止：

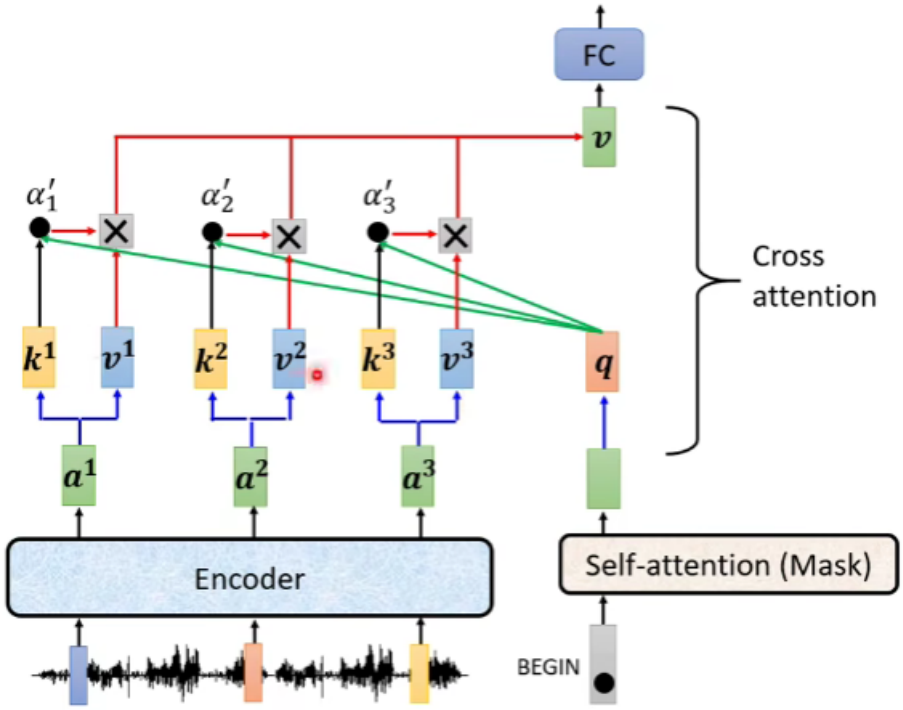


3 Encoder 和 Decoder 相连接

q来自于Decoder，k和v来自于Encoder：

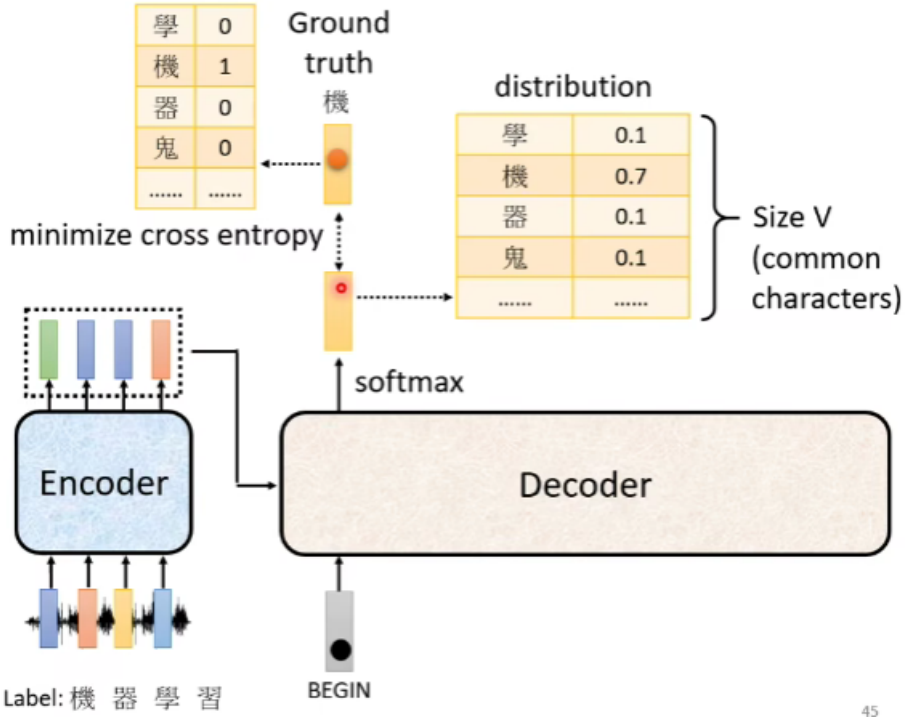


具体计算过程如下：



4 训练过程

以语音识别为例，输入为一串声音信息，输出为中文字符。输出实际上是以概率的形式表示的，我们希望某一个词的概率最大为1，其他的都为0，只需要将输出的结果与预期相减，得到损失函数，最小化这个损失函数即可：


學 0
機 1
器 0
鬼 0
....
Ground truth
機
distribution
學 0.1
機 0.7
器 0.1
鬼 0.1
....
Size V (common characters)
minimize cross entropy
softmax
Encoder
Decoder
BEGIN

Label: 機 器 學 習