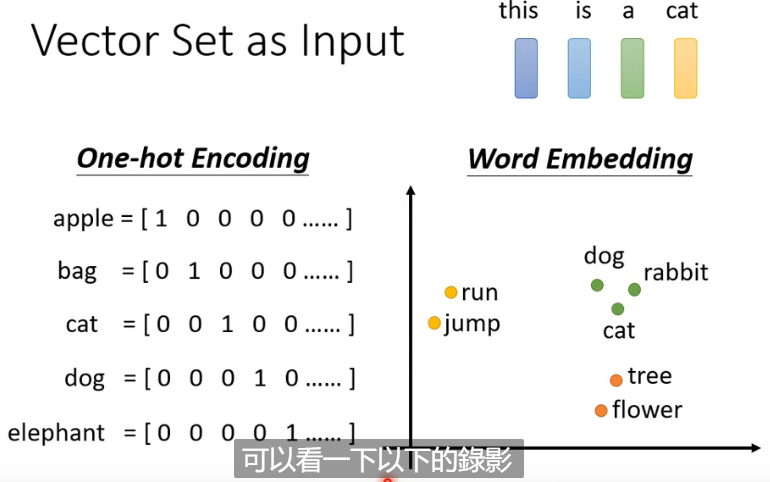
将词表示成向量

One-hot Encoding

通过一个很长的向量其中每一个词对应一个位置的值设为1，使得每一个维度对应一个词。但这会使得每一个词之间都看不出它们之间的关系。

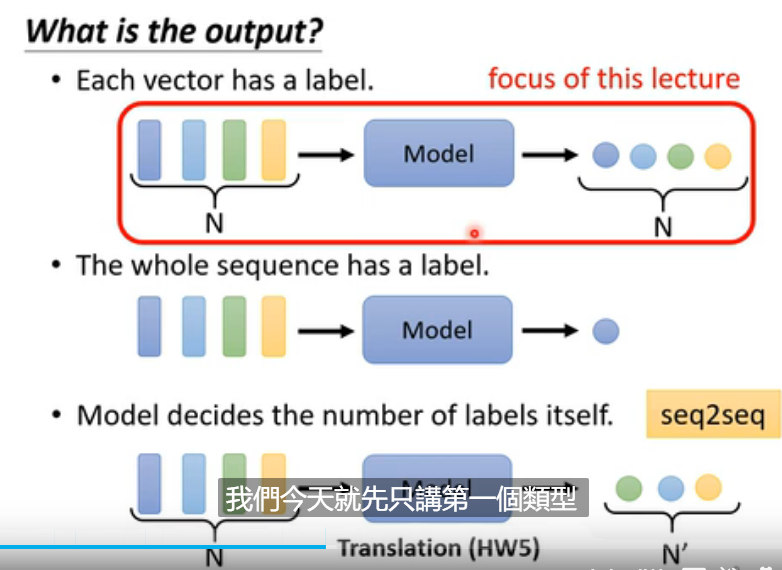
Word Embedding

将每个词对应着一个向量。



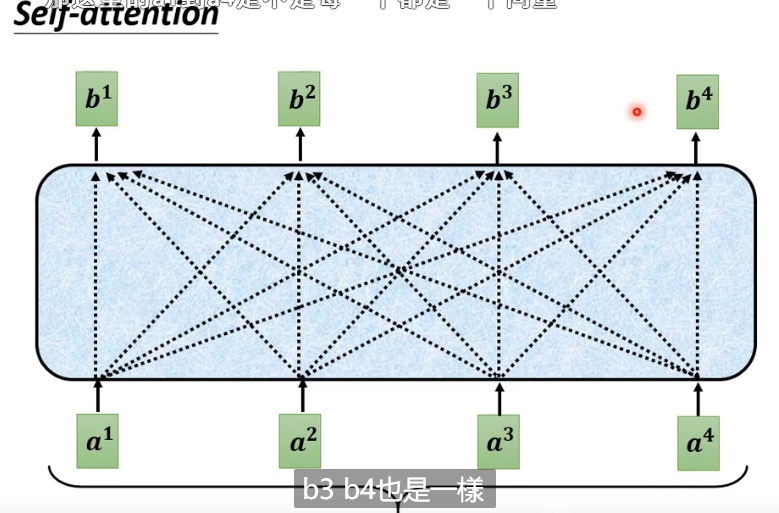
模型的输入为一堆向量，这些向量可以是语音，图，句子

输出就有三种可能性，每一个向量都对应一个标签、整个输入对应着一个输出结果、多个输出对应着多个输出。

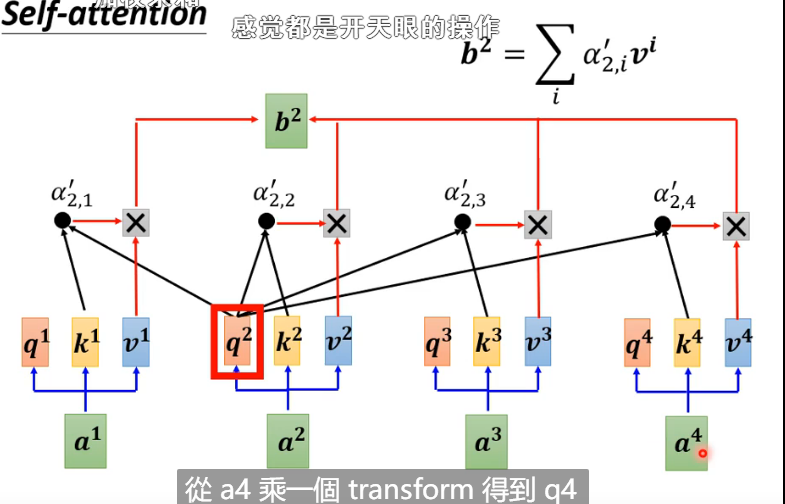


Self-attention

自注意力机制的输入可能是未处理过的input也可能是某个隐藏层的输出，都是一排的向量，通过考虑计算这一排所有的向量之后得到对应的输出。

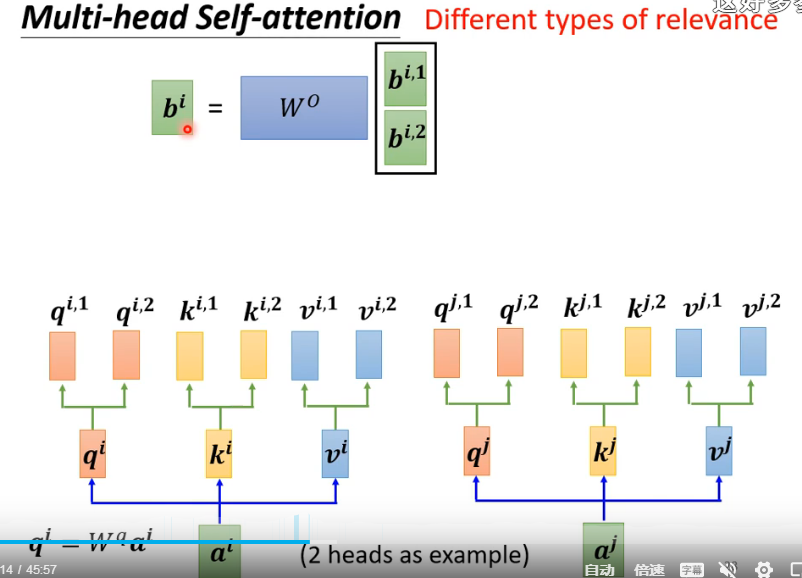


通过计算每个向量相乘不同的矩阵w得到q和k，然后将q和k相乘得到两个向量的相关性。当一个向量和其他向量包括自己算完相关性后，通过一层softmax将这些值归一化，通过计算w和向量相乘，得到v，然后再和归一化后的相关性的值相乘，然后全部加强了就得到了输出结果



Multi-head Self-attention

通过w得到q，k，v后再乘多个矩阵得到多个不同的q，k，v，并且q，k，v的数量要相同并且对应，然后同一类的q，k，v进行注意力机制的运算。最后将所有结果进行拼接再乘上一个矩阵就得到最后的结果。

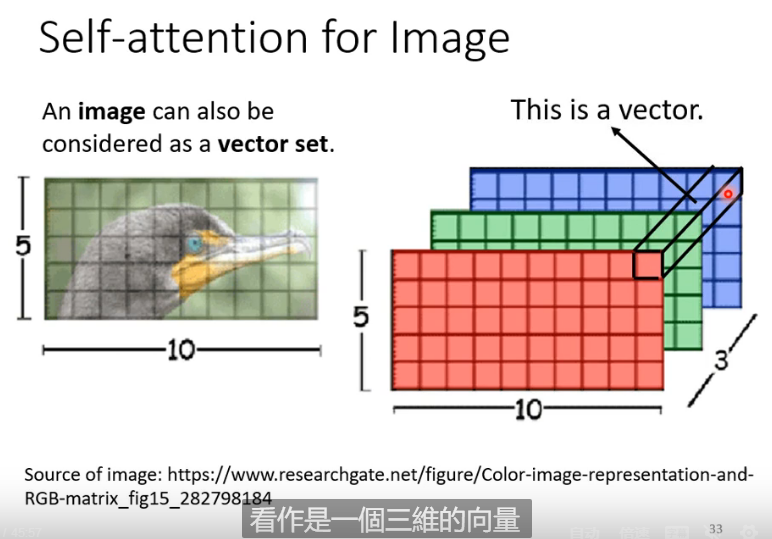


Positional Encoding

通过位置编码解决在注意力机制计算中无法区别每一个词之间的距离远近无法区别的问题。

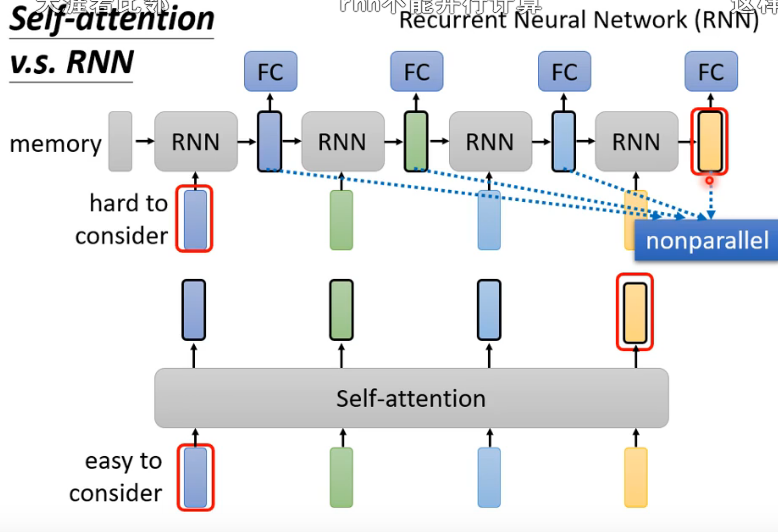
Self-attention for image

将图片看成一个多维的向量，然后就可以通过self-attention来处理图片。



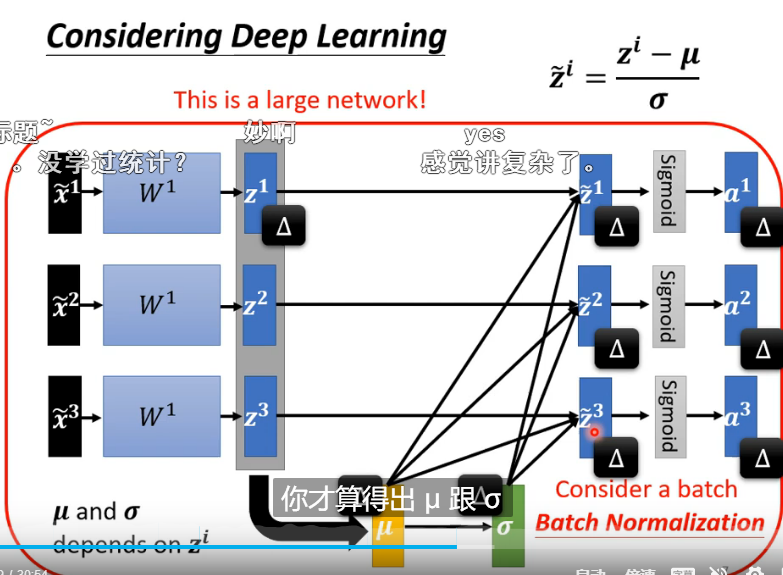
Rnn

在通过神经网络的输入多了一个state，这个state是保存的上一个向量输入的内容，每一次的输出都会有一个state。因此Rnn是都有考虑了上一个的输出的情况，而self-attention则是考虑的整个句子的输入。

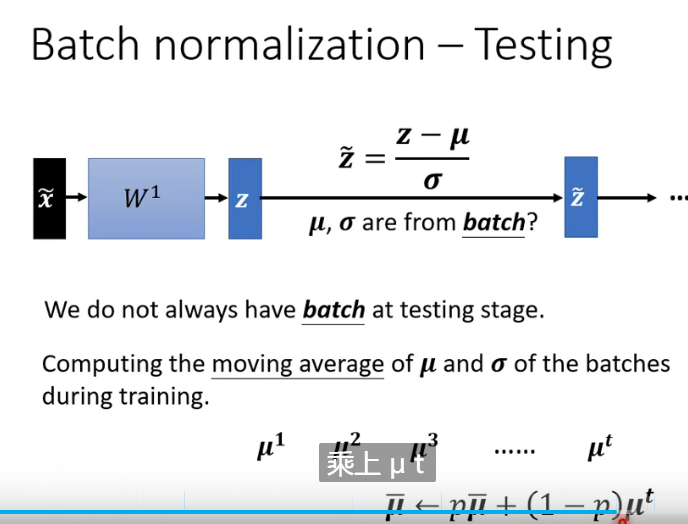


Batchnormalization

将所有的数据中同一个维度的值取出来，计算出每一个维度的均值和标准差，然后将每一个维度减去该维度的均值再除标准差进行标准化，使得每一个维度都服从均值为0方差为1的分布。

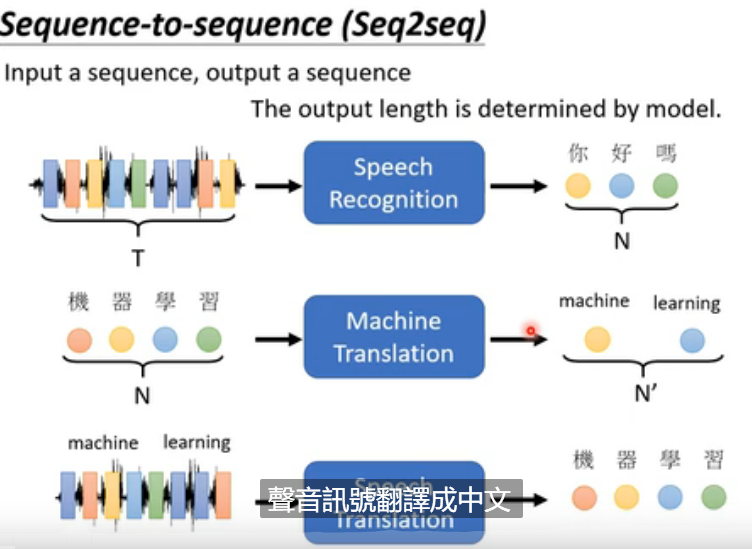


在测试的时候，由于有可能的输入就只有一个数据，无法做标准化，因此pytorch中自动完成了将训练的时候的均值和方差累计统计出来。因此在测试的时候只需要将输入的数据减去统计出来的均值除以统计出来的方差就可以了。



Seq2seq

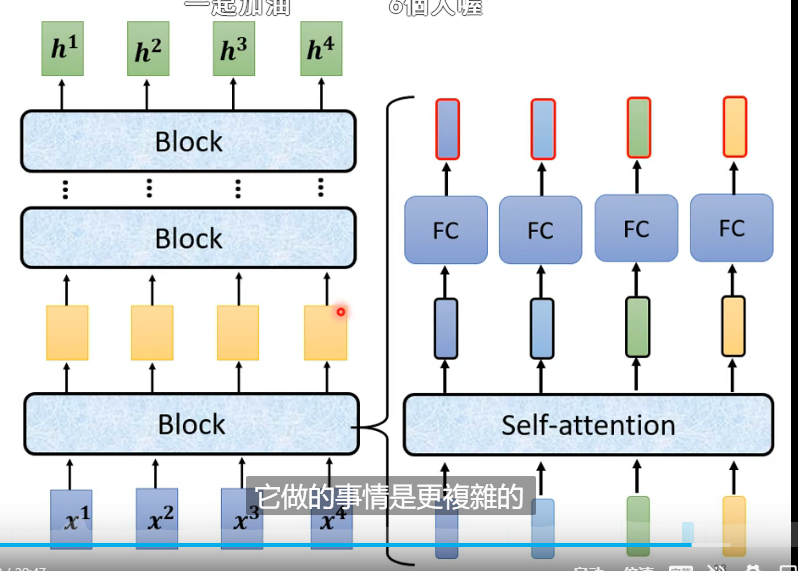
通过输入一段句子然后输出另一段句子，其中句子长度不一样，输出句子长度是由机器决定。

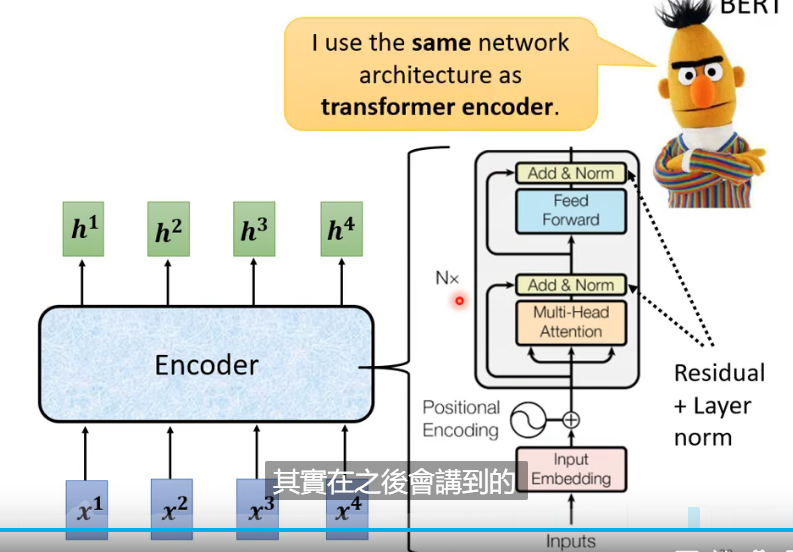


Seq2seq分为两个模块，分别为encoder和decoder

Encoder要做的事情就是给一排的向量输出另外一排的向量，其中会有很多的block。

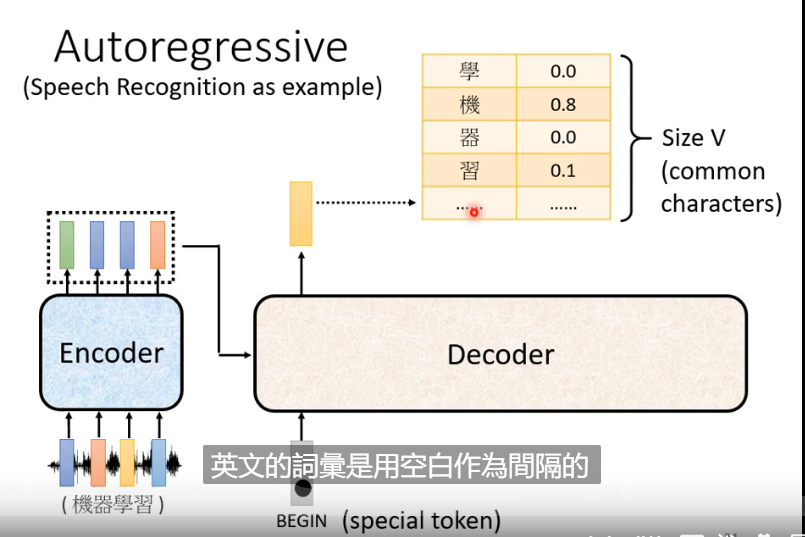
在transformer中encoder用到的self-attention，每一个block中做self-attention得到输出，然后再通过一层全连接层，最后得到的输出就是一个block的输出，其中还用到了残差网络





Decoder

通过输入一个向量，进入decoder，decoder将encoder的输出和输入得到一个输出。



Mask-attention

只考虑当前已知的输出结果，忽略未知的结果。

Cross attention

其中attention的q和k是中encoder的输入，v则是之前的mask-attention的输出。因此cross attention是连接encoder和decoder的桥梁。