# 1.Seq2Seq

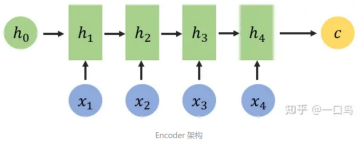
传统的RNN模型只能解决序列到序列的等长度序列预测问题，而seq2seq通过引用编码器（Encoder）和解码器（Decoder）能够解决不等长的序列预测问题。通常 Encoder 的架构较为固定，而 Decoder 的架构较为多样，区别主要在于编码后的语义向量被运用在解码时的具体方式。

Seq2Seq的结构分为：编码器部分（Encoder）和解码器部分（Decoder）。

## 1.模型结构

### Encoder

在 Encoder 中，输入为一个按输入时刻展开的序列，序列中每个位置的输入对应到不同时刻的输入上。对于第t时刻而言，模型的输入为当前时刻的序列内容及上一时刻的隐含层（也称为记忆单元）输出。因此，当前时刻的输出既考虑了当前时刻的信息，还保留了历史时刻存留的信息。具体架构如下：



Encoder 架构

用数学公式表示，对于时刻t，隐藏层$h_t$可以表示为：

$$h_t = tanh(W_{xh}x_t + b_{xh} + W_{hh}h_{t+1} + b_{hh})$$

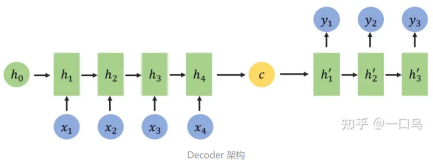
t时刻的输出结果为：

$$y = softmax(W_{ho}h_t + b_{ho})$$

### Decoder

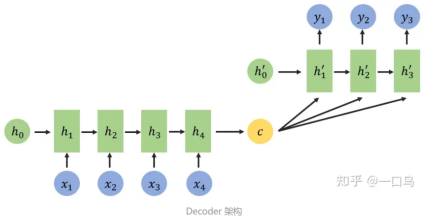
在 Encoder 中，我们得到了源序列的语义信息，并以语义向量c的形式保留了下来。接下来我们就需要通过解码器（Decoder）对语义向量c进行解码，根据解码方式的不同，Dercoder分为以下几种形式：

1）语义向量c仅作为初始状态，解码器后面时刻的运算与c无关

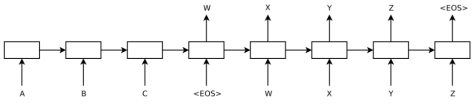


Decoder 架构

2）语义向量c作为解码器每个时刻的输入




3）Teach Forcing




训练过程中将解码器的输入进行平移过后作为解码器的输出标签，这样做（利用当前token和下一时刻token有强烈的关系来预测下一时刻token，这会加快训练速度，提高模型训练效果）。推理过程中，将上个序列的输出作为当前序列的输入。

**缺点：**1.编码器无法提取位置信息；2.对于长语句，由于语义向量c维度的限制，可能导致效果不好

# 2.应用场景

（1）机器翻译：Seq2Seq最经典的应用，当前著名的Google翻译就是完全基于Seq2Seq+Attention机制开发的。

（2）文本摘要自动生成：输入一段文本，输出这段文本的摘要序列。

（3）聊天机器人：第二个经典应用，输入一段文本，输出一段文本作为回答。

（4）语音识别：输入语音信号序列，输出文本序列。

（5）阅读理解：将输入的文章和问题分别编码，再对其解码得到问题的答案。

（6）图片描述自动生成：自动生成图片的描述。

（7）机器写诗歌，代码补全，故事风格改写，生成commit message等。