Seq2seq

Seq2seq 思想：

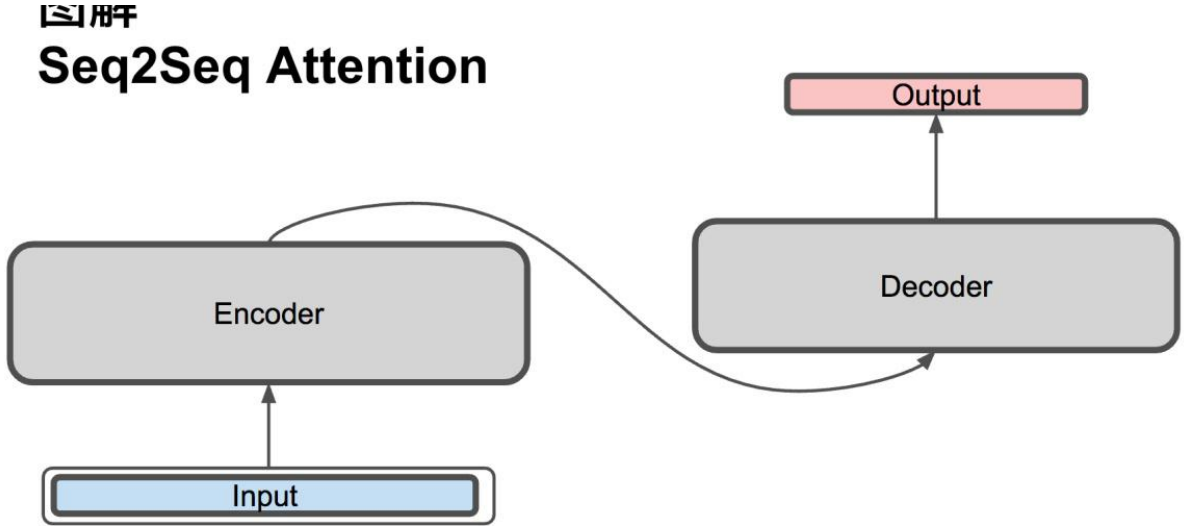
Seq2seq 是 encoder-decoder 框架的一个算法，举例使用 rnn 做 encoder 和用 rnn 来做 decoder。

解决的问题是：

不定长输入到不定长输出的问题。

将不定长词序列转化为定成向量表示这句话，然后解码为不定长序列。

大框架：



我理解 seq2seq 时碰到的几个问题：

1. 模型训练过程中到底有多少个矩阵
2. encoder 是一个 rnn，decoder 是一个 rnn
3. encoder 的初始值 net1 怎么定，decoder 的初始隐向量 net1 怎么定？
4. 最终 decoder 解码的时候输出层是一个很大的词表，如果使用 softmax 的话，计算时间复杂度是不是很高？没用优化算法吗？

下面是整体流程：

理解的时候比较费力的点：

1）attention 怎么计算？思路如下很清晰

attention 的计算方法有多种（在求 score 那块）

思路：简单的点积方法，或者是去学习一个矩阵（第二种方法）然后计算 score，或者学习更多的参数来计算 score

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

输入： $x = (x_1, \ldots, x_{T_x})$

输出： $y = (y_1, \ldots, y_{T_y})$

(1) $h_t = RNN_{enc}(x_t, h_{t-1})$, Encoder方面接受的是每一个单词word embedding，和上一个时间点的hidden state。输出的是这个时间点的hidden state。

(2) $s_t = RNN_{dec}(\hat{y_{t-1}}, s_{t-1})$ ， Decoder方面接受的是目标句子里单词的word embedding，和上一个时间点的hidden state。

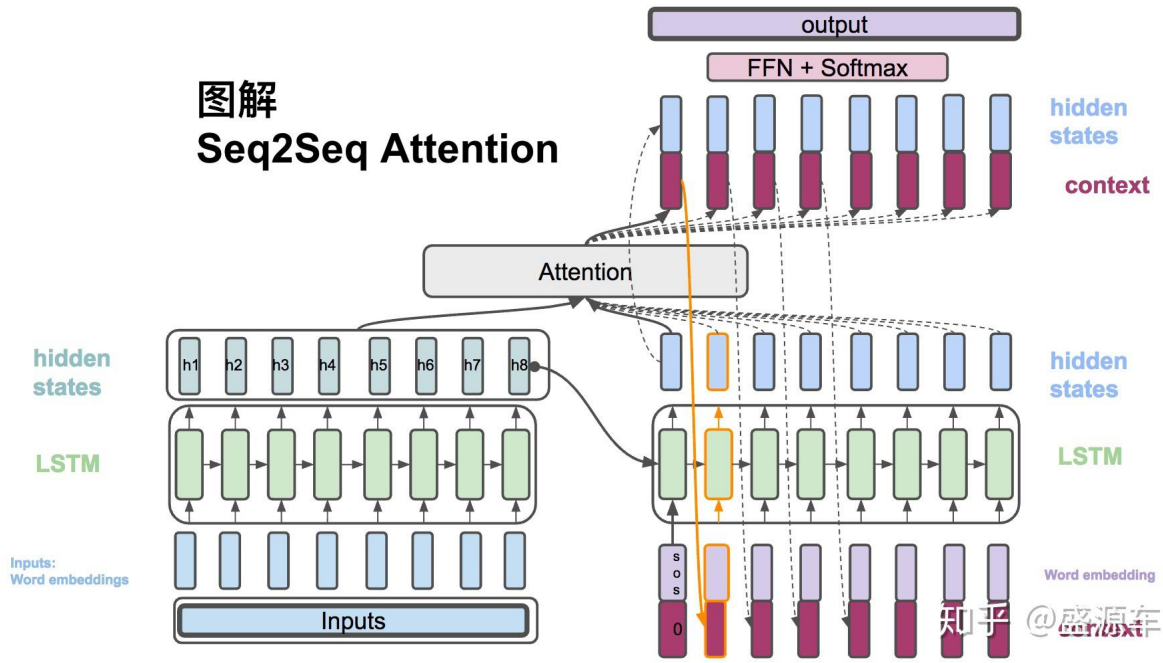
(3) $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$, context vector是一个对于encoder输出的hidden states的一个加权平均。

(4) $\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$, 每一个encoder的hidden states对应的权重。

(5) $e_{ij} = score(s_i, h_j)$, 通过decoder的hidden states加上encoder的hidden states来计算一个分数，用于计算权重(4)

(6) $\hat{s_t} = tanh(W_c[c_t; s_t])$, 将context vector 和 decoder的hidden states 串起来。

(7) $p(y_t|y_{<t}, x) = softmax(W_s \hat{s_t})$ ， 计算最后的输出概率。



详解上面的图：

1. encoder 的 rnn 和 decoder 的 rnn 是两个独立的 rnn，他们俩的参数可以看成的独立的

2. 两个 rnn 通过 attention 间接的连接在一起，attention 之后得到的 ct 和 st 做拼接，其中 st 由 s(t-1)和相对应的前一个 y(t-1)预测值通过一个 rnn cell 获得, 这一点跟 encoder 不一样, encoder 使用的是 xt。而 y(t-1)预测值是 st-1 和 c(t-1)拼接之后经过全联接，然后 softmax

分类而成。

Seq2seq 比我想象的要复杂，主要在解码那一块。

Seq2seq 经常会使用 beam search 做优化。因为如果第一个预测的词预测错了或者中间的某一个词预测错了，后面的词就可能会全错。
Beam search 的思想就是每次预测 topk 个词。
如果每次都预测 topk 个词。

Beam search 在模型训练中不需要使用，因为在模型训练中，预测下一个词的时候会用正确的词进行预测。
Beam search 在 seq2seq 中的解码中的应用：
1）在解码第一个词的时候，softmax 选择 topk 的词。
2）在解码第二个词的时候，这 topk 的每个词都会拿来预测第二个词，每个词都产生 topk 个候选词。这个时候在第二个词的时候就有 topk^2 个词作为候选。
3）减枝思想。从 topk^2 个词中选择 topk 的值作为第二个词的候选词
4）重复 2）3）即可。
5）在解码到最后一个词的时候选择 top1