Attention 机制的 Seq2Seq 模型。模型主要包括 Encoder 与 Decoder 两部分组成。

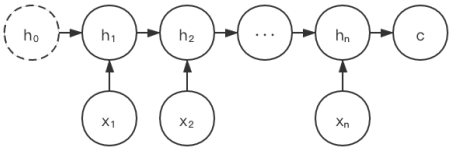

图 3 Encoder 框架

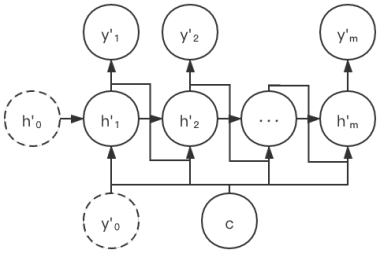

图 4 Decoder 框架

3.1 编码器 Encoder

Encoder 与一般的 RNN 类似，但在中间神经元中没有输出。下图是 Encoder 部分，Encoder 的 RNN 接受输入 x ，最终输出一个编码所有信息的上下文向量 c ，中间的神经元没有输出。Decoder 主要传入的是上下文向量 c ，然后解码出需要的信息。

其中的上下文向量c可以采用如下列举的常见方式进行计算。

$$c = h_N \quad (1)$$

$$c = q(h_N) \quad (2)$$

$$c = q(h_1, h_2, \ldots, h_N) \quad (3)$$

上述方式一中上下文向量c直接由最后一个神经元的隐藏状态 h_N 表示；方式二中上下文向量 c 由最后一个神经元的隐藏状态上进行某种变换 h_N 而得到， q 函数表示某种变换；方式三种上下文向量c由所有神经元的隐藏状态 $h_1, h_2, \ldots, h_N$ 计算得到。得到上下文向量 c 之后，需要传递到 Decoder。

3.2 解码器 Decoder

Decoder 有许多不同结构，下图为本方案采用的 Decoder 框架。

具有自己的初始隐藏层状态 h'_0 ，而每一个神经元的输入为将上一个神经元的输出 y' 与上下文向量 c （由 Encoder 编码后传入）。对于第一个神经元的输入 y'_0 ，通常是句子其实标志位的 embedding 向量。第三种 Decoder 的隐藏层及输出计算公式：

$$h'_t = \sigma(Uc + Wh'_{t-1} + Vy'_{t-1} + b) \quad (4)$$

$$y'_t = \sigma(Vh'_t + c) \quad (5)$$

3.3 Attention 机制

Attention 即注意力机制，是一种将模型的权重放在当前信息编码上的一种机制，通过使用权重进行相关计算实现。在 Attention 机制下，Decoder 的输入不再

是固定的上下文向量 c ，而对根据当前进行的神经元计算当前所需的 c 。

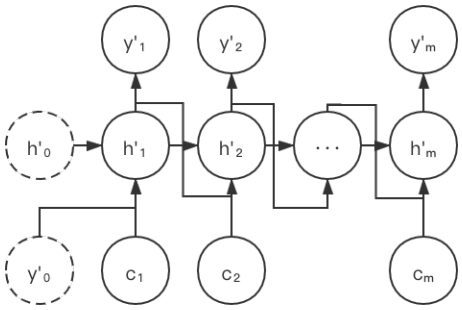

图 5 Attention 框架

Attention 机制下 需要保留 Encoder 每一个神经元的隐藏层向量 h ，然后 Decoder 的第 t 个神经元要根据上一个神经元的隐藏层向量 h'_{t-1} 计算出当前状态与 Encoder 每一个神经元的相关性 e_t 。

$$e_t = [a(h'_{t-1}, h_1), a(h'_{t-1}, h_2), \ldots, a(h'_{t-1}, h_N)] \quad (6)$$

e_t 是一个 N 维的向量 (Encoder 神经元个数为 N)，若 e_t 的第 i 维越大，则说明当前节点与 Encoder 第 i 个神经元的相关性越大。 e_t 的计算方法有很多种，即相关性系数的计算函数 a 有很多种：

$$a(h'_{t-1}, h_i) = h_i^T h'_{t-1} \quad (7)$$

$$a(h'_{t-1}, h_i) = h_i^T W h'_{t-1} \quad (8)$$

$$a(h'_{t-1}, h_i) = \tanh(W_1 h_i + W_2 h'_{t-1}) \quad (9)$$

此方案我们选择第 2 种(8)，上面得到相关性向量 e_t 后，需要进行归一化，使用 softmax 归一化。然后用归一化后的系数融合 Encoder 的多个隐藏层向量得到 Decoder 当前神经元的上下文向量 c_t ：

$$\alpha_t = softmax(e_t) \quad (10)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{N} \exp(e_{tj})} \quad (11)$$

$$c_t = \sum_{i=1}^{N} \alpha_{ti} h_i \quad (12)$$