

Memory network

1. 记忆神经网络在 QA 方面的应用。传统的深度学习模型（RNN，LSTM 等）使用 hidden states 或者 attention 机制作为他们的记忆功能，但是这种方法产生的记忆太小了，无法精确记忆一段话中表达的全部内容。
2. （博客知识）下面介绍一个模型，模型主要包含一系列的内存单元（可以看成是一个数组，每个元素保存一句话的记忆）和 I，G，O，R 四个模块。

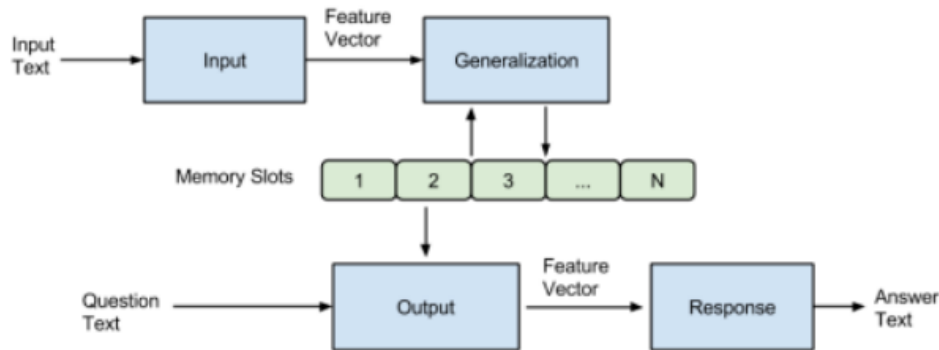


Figure 3: Memory Network architecture.

简单介绍：

1. 输入的文本经过 Input 模块编码成向量，然后将其作为 Generalization 模块的输入。
 2. 该模块根据输入的向量对 memory 进行读写操作，即对记忆进行更新。
 3. 然后 Output 模块会根据 Question（也会经过 Input 模块进行编码）对 memory 的内容进行权重处理，将记忆按照与 Question 的相关程度进行组合得到输出向量，最终 Response 模块根据输出向量编码生成一个自然语言的答案出来。
- 每个模块的作用如下：

1. Convert x to an internal feature representation $I(x)$.
2. Update memories \mathbf{m}_i given the new input: $\mathbf{m}_i = G(\mathbf{m}_i, I(x), \mathbf{m}), \forall i$.
3. Compute output features o given the new input and the memory: $o = O(I(x), \mathbf{m})$.
4. Finally, decode output features o to give the final response: $r = R(o)$.

Model 细节

1. input 输入是一个简单的 embedding lookup 操作，将原始文本转化为词向量。
 2. G 模块直接将输入的向量存储在 memory 数组的下一个位置。
- 主要的工作在 O 和 R 两个模块进行：
3. O 模块根据问题向量，搜索以前所有的向量，然后找出与他最相关的向量。公

$$o_1 = O_1(x, \mathbf{m}) = \arg \max_{i=1, \dots, N} s_O(x, \mathbf{m}_i)$$

式为：

然后根据选出的 O_1 和输入 x 选出与他们两个最相关的记忆 O_2 .

$$o_2 = O_2(x, \mathbf{m}) = \arg \max_{i=1, \dots, N} s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_i)$$

这样就选出了与 question 最相关的记忆。然后就作为 R 模块的输入，用于生成

$$r = \operatorname{argmax}_{w \in W} s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], w)$$

最终答案。

上述中评分函数 s_O 和 s_R 都有一种形式，即嵌入式模型，所以可以携程下面的函数：

$$s(x, y) = \Phi_x(x)^\top U^\top U \Phi_y(y).$$

4. 损失函数的选择：模型选择 margin ranking loss 作为损失函数，即要求正确答案的得分比错误答案的得分高至少一个 margin r 。

$$\begin{aligned} & \sum_{\bar{f} \neq \mathbf{m}_{o_1}} \max(0, \gamma - s_O(x, \mathbf{m}_{o_1}) + s_O(x, \bar{f})) + \\ & \sum_{\bar{f}' \neq \mathbf{m}_{o_2}} \max(0, \gamma - s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_{o_2}) + s_O([x, \mathbf{m}_{o_1}], \bar{f}')) + \\ & \sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], r) + s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \bar{r})) \end{aligned}$$

没看懂这个公式。

总结：主要是讲了 memory network 的工作原理，以及使用的一些公式介绍。

大致就是（QA 举例子）把一段话变成词向量，按顺序存入数组中，提出问题同样转化为词向量找到与他最相似的记忆，作为答案。

心得：对一些公式的具体运算流程不太熟，只知道怎么用。最近考试有点多，打算找个例子实验一下，还没有时间找。