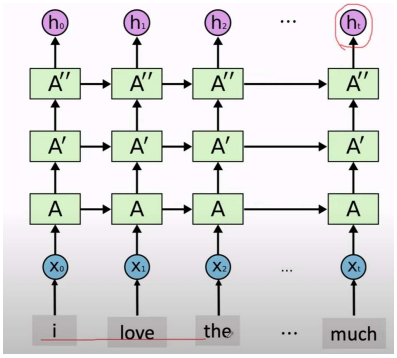
Making RNNs more effective

2021年2月4日 14:24

1. Stacked RNN (多层RNN)

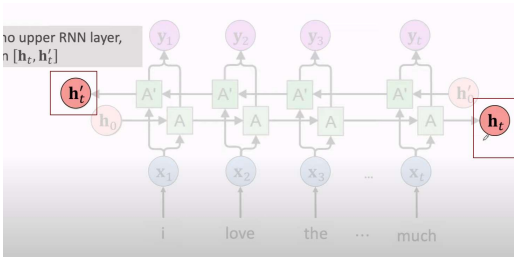
- 链接：https://www.youtube.com/watch?v=pzWHk_M23a0



前面两个模型的最主要的问题：如果训练的数据集不够，而多层lstm叠加让embedding层的参数过多，就会出现过拟合的现象，导致实验结果不理想

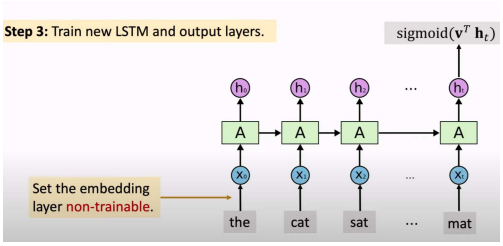
1. 最底层的是每个单词，x_i 为每个单词对应的词向量
2. 每层的lstm都有两个hidden vector h，被copy成两份，一份属于当前lstm层，另外一份被传递到下一层
3. 最后h_t 承载了整个句子的信息
4. 在写代码的时候，设置return_sequences = True便能使得多层模型堆叠。
5. 如果训练数据足够的多，多层RNN效果要比单层RNN好的多

2. Bidirectional RNN (双向RNN)



1. 双向RNN分为从后往前和从前往后两个方向，两个方不共享参数，也不共享状态，参数的个数是单向的double
2. 如果是多层的话，y_i 就会是下一层的隐藏参数传递到下一层
3. 最后双向RNN返回两个状态向量h_t ，双向RNN的返回值就是两个状态向量的结合，维度直接相加!
4. 双向的效果在数据两充足的前提下比单向的好

3. Pretrain the embedding layer



1. 在大的数据集上训练模型，然后保留训练模型的embedding层
2. 与训练的任务和目标任务之间的关系越相似，与训练的效果就会越好
3. 与训练的神经网络是任何的网络结构都行，只要有embedding层就行！！
4. 训练好之后，把与训练模型上面的层丢掉，只保留embedding层和参数，然后搭建我们自己的RNN网络
5. 固定预训练的embedding层，训训练我们新的RNN 网络