

循环神经网络(RNN)入门帖：向量到序列，序列到序列，双向RNN，马尔科夫化

原创 David9 机器学习算法与自然语言处理 2018-04-05



作者: [David 9](#)

地址: <http://nooverfit.com/>

rnn似乎更擅长信息的保存和更新，而cnn似乎更擅长精确的特征提取；rnn输入输出尺寸灵活，而cnn尺寸相对刻板。

1 发问

聊到循环神经网络**RNN**，我们第一反应可能是：**时间序列**（time sequence）。

确实，RNN擅长时间相关的应用（**自然语言，视频识别，音频分析**）。但为什么**CNN**不容易处理时间序列而**RNN**可以？为什么我们之前说过RNN有一定的记忆能力？

2 普通预测

数学上，如果我们想要预测一个单词 x 的后一个单词 y ，我们需要3个主要元素（输入单词 x ； x 的上下文状态 h_1 ；通过 x 和 h_1 输出下一个单词的函数比如softmax）：



来自：<http://suriyadeepan.github.io/2017-01-07-unfolding-rnn/>

数学计算如下：

$$h1 = \tanh (W1x+b1)$$

$$o = \tanh (W2h1+b2)$$

$$y = \text{softmax} (o)$$

上面是一个很简单的有向无环图（DAG），但是，这只是一个时刻t 的单词预测，这种简单的预测甚至可以用cnn或者其他简单预测模型替代。

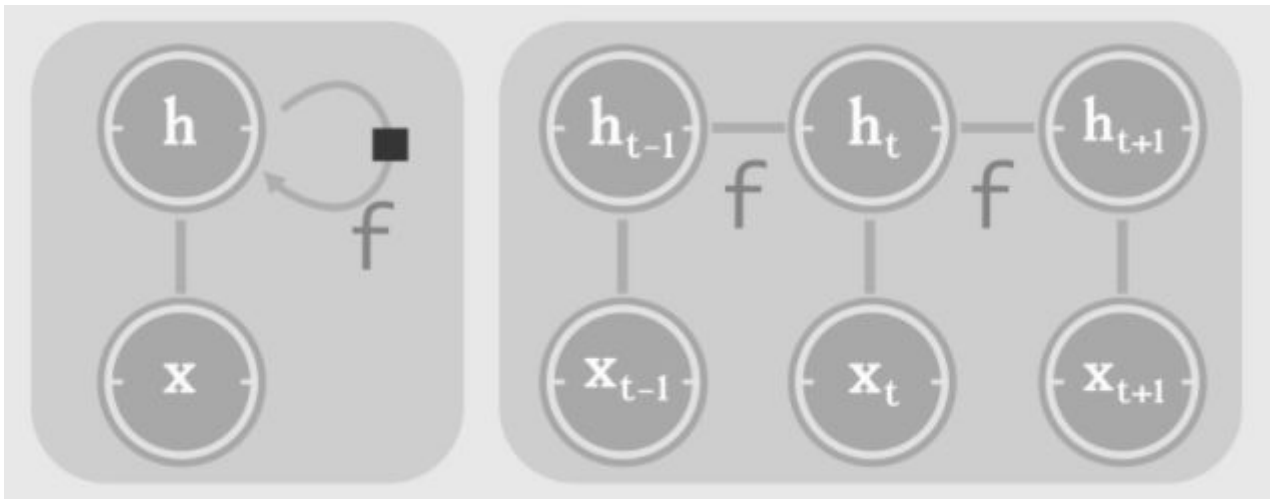
3 RNN的引入

然而，cnn对于**更新状态**或者**保存状态**却并不擅长，我们知道，下一个时间点t+1，单词x的上下文（状态）就改变了：



因此，RNN的**门限结构**和CNN**卷积结构**的不同（信息保存方式的不同）也一定程度导致RNN擅长处理**时间序列**的问题。即使我们不用**门限网络**而用其他模型，我们也需要类似上图的循环结构，**把上下文状态在每一个时间点进行更新，并保存下来。**

所以，在**时间序列**的应用中，更新每个时间点的状态是如此重要，我们需要rnn这样的网络：



在每个时间点，都使用同样的更新函数f 更新上下文状态，每个时间点t的状态都是基于上一个时间点t-1的状态和本次信号 x_t 的输入：

$$h_t = f(h_{t-1}, x_t; \theta)$$

另外，RNN的门限网络有天然的**马尔科夫化**的性质，当前的**状态S3**经过多次循环已经包含了几个时间点以前的状态信息（其中分号代表用参数 θ 编码前面状态）：

$$s_3 = f(s_2; \theta) = f(f(s_1; \theta); \theta)$$

当前的预测只需要根据当前的状态进行预测。**这种巨大的保存状态信息的能力似乎正是RNN门限单元擅长的。（cnn似乎更擅长精确的特征提取）**

传统的rnn是每个时刻t，输入一个单词，生成另一个单词，而实际情况并不都是这样简单。

4 RNN变形

最后，我们看一些变形的RNN结构。

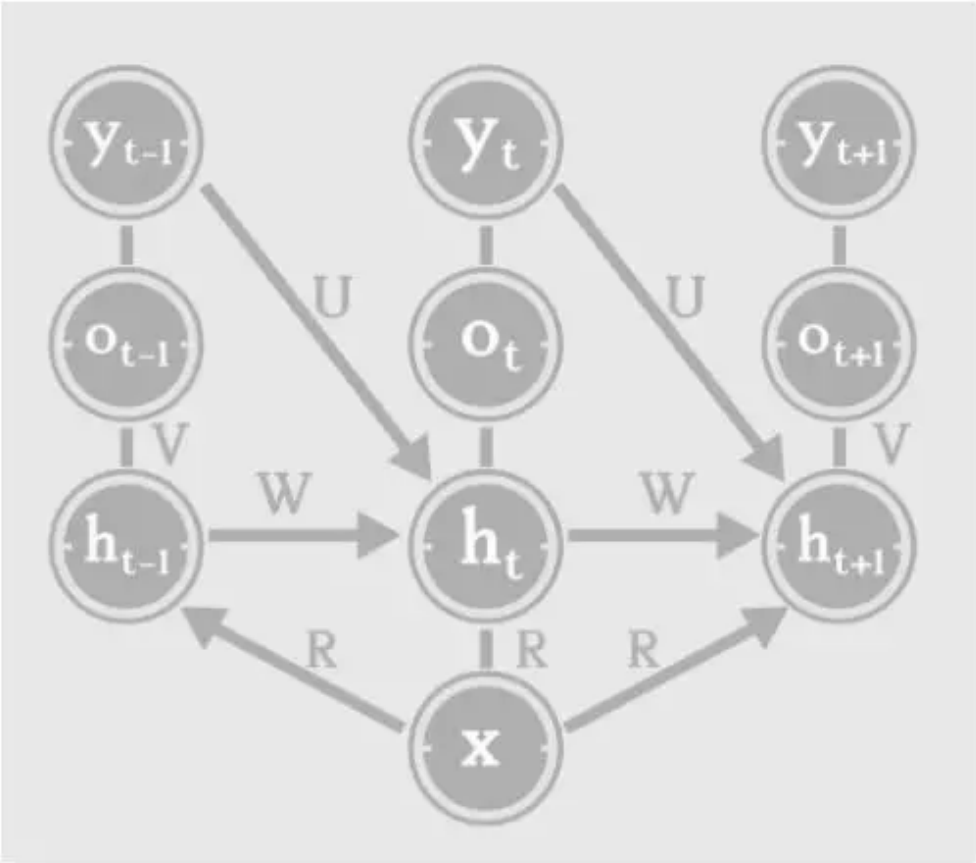
向量到序列 (Vector to Sequence)

有一些应用如根据一张图片，输出图片的字幕：



这种问题的RNN，需要输入图片的特征向量x（比如cnn的最后一层隐层），输出是一句话的字幕，但是这句话y的单词是一个一个生成的。

这就要求rnn从单个向量中，一次性生成一个时间序列：



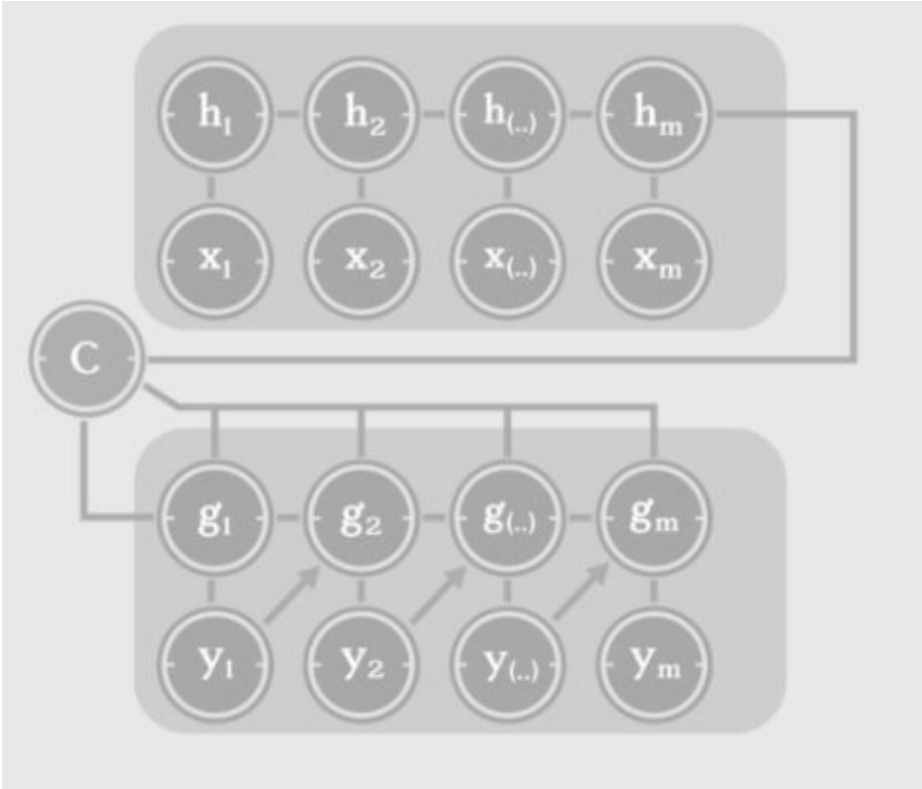
来自：<http://suriyadeepan.github.io/2017-01-07-unfolding-rnn/>

当然，时间序列的生成遵循了按照时间顺序循环更新内部状态的规则。

5 序列到序列 (seq2seq)

机器翻译问题中，比如一句英语句子翻译成法语句子，不一定对应的单词数量是相等的。所以传统的 rnn 一定需要修改。

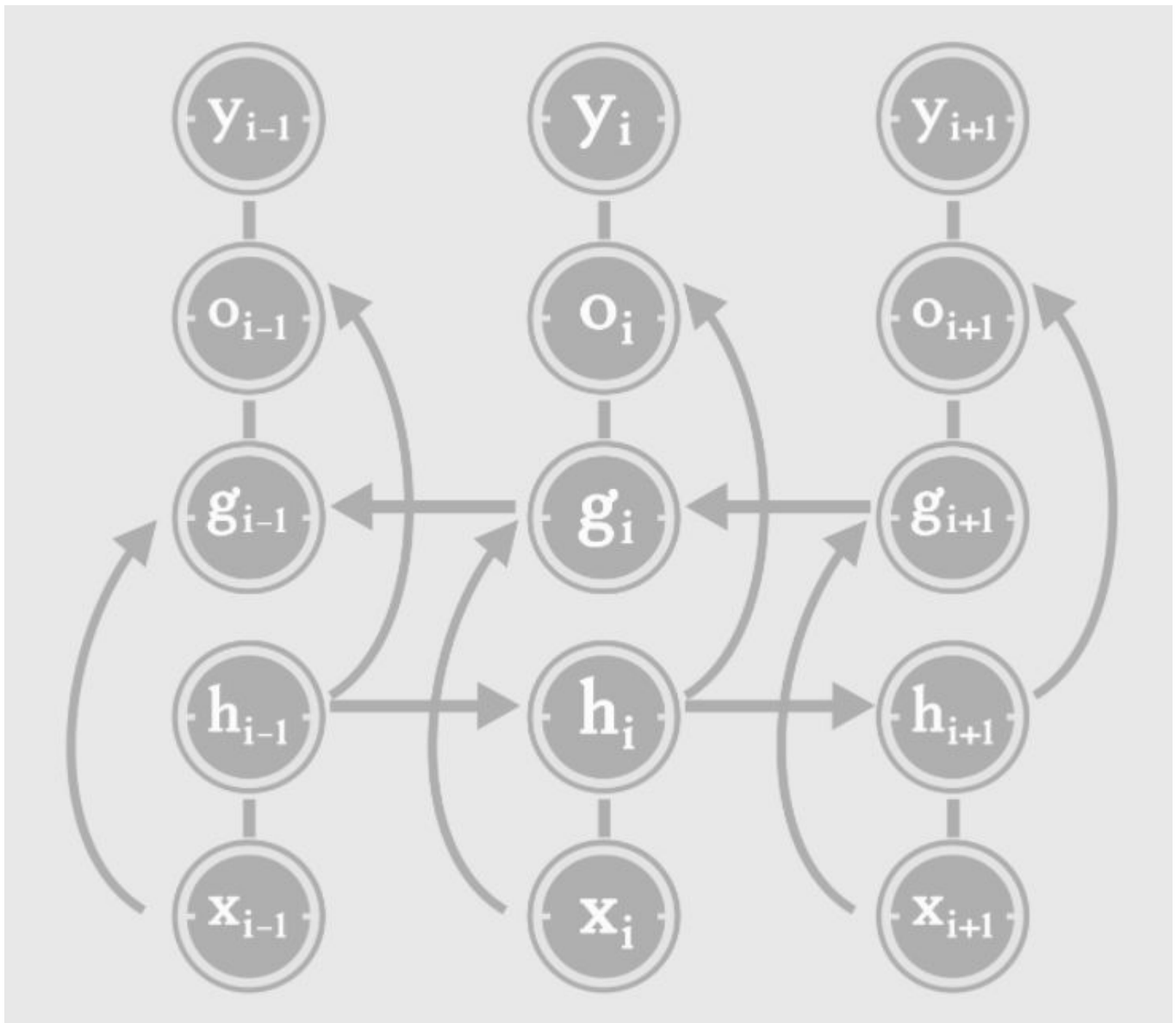
常见的做法是用**两个 rnn**，一个 rnn 用来**编码**句子 (**encoder**)，一个 rnn 用来**解码**成想要的语言 (**decoder**)：



这里的C 是上下文的信息，和编码好的隐层信息一起，送入decoder的输入，进行翻译。

6 双向RNN：

传统的RNN预测一个单词，只是捕捉这个单词之前的上下文状态，而双向RNN还捕捉了这个单词后面的单词环境对该词的影响：



上图的双向RNN其实可以看做两个RNN，一个RNN就是我们之前提到的传统RNN（只有隐藏状态h那一层）；还有一个RNN是捕捉单词之后的单词环境的RNN（隐藏状态g那一层）。

这样的RNN把一个单词向左和向右的上下文环境信息考虑进去，准确率一般会有所提高。

参考:<http://suriyadeepan.github.io/2017-01-07-unfolding-rnn/>

END

推荐阅读：

[哈工大SCIR中心推出对话技术平台-DTP（测试版）](#)

[【深度学习实战】pytorch中如何处理RNN输入变长序列padding](#)

[【机器学习基本理论】详解最大后验概率估计（MAP）的理解](#)