

092 | 序列建模的深度学习利器：RNN基础架构

2018-05-04 洪亮劼

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 06:57 大小 3.19M



前面我们介绍了一个重要的文本模型，Word2Vec，我们聊了这个模型的基本假设，模型实现，一些重要的扩展，以及其在自然语言处理各个领域的应用。

接下来，我们来讨论更加复杂的**基于深度学习的文本分析模型**。这些模型的一大特点就是更加丰富地利用了**文字的序列信息**，从而能够对文本进行大规模建模。

今天，我们首先来看一看，序列建模的深度学习利器 **RNN**（Recurrent Neural Network，递归神经网络）的基本架构。

文本信息中的序列数据

我们在之前介绍 Word2Vec 的时候，讲了为什么希望能够把上下文信息给融入到模型当中去。一个非常重要的原因，就是在最早的利用“词包”（Bag of Word）的形式下，离散的词向量无法表达更多的语义信息。那么，从文本的角度来讲，很多研究人员都面对的困扰是，如何对有序列信息的文本进行有效的建模？同时，对于广大文本挖掘的科研工作者来说，这也是大家心中一直深信不疑的一个假设，那就是**对文字的深层次的理解一定是建立在对序列、对上下文的建模之中**。

你可能有一个疑问，文字信息中真的有那么多序列数据吗？

其实，从最简单的语义单元“句子”出发，到“段落”，到“章节”，再到整个“文章”。这些文字的组成部分都依赖于对更小单元的序列组合。例如，句子就是词语的序列，段落就是句子的序列，章节就是段落的序列等等。不仅是“词包假设”无法对这样的序列进行建模，就算是我们之前提到的 Word2Vec 等一系列学习词向量或者段落向量的方法，也仅仅能考虑到一部分的上下文信息。

还有更加复杂的文字序列，比如对话。人与人的对话很明显是有顺序的。两个人之间进行对话，当前所说的字句都是根据对方的回应以及整个对话的上下文所做出的选择。如果要对这样复杂的文字序列进行建模，传统的不考虑序列的模型方法是肯定不能奏效的。

那么，传统的机器学习领域，有没有能够对时序信息建模的工具或者模型呢？

传统机器学习中的序列模型

在传统的机器学习领域当然有不少模型是针对序列进行建模的。最经典的要数“**隐马尔科夫模型**”（Hidden Markov Model），有时候又简称为 **HMM**。在比较长的一段时间里，HMM 都是常用的对序列建模的缺省（Default）模型。我们今天的分享不是专门针对 HMM 模型，但是对 HMM 的一个简单介绍，有助于我们理解为什么需要 RNN。

HMM 的一个最基本的假设是：当前的序列数据是根据一些隐含的状态产生的。具体来说，HMM 的架构是这样的。我们认为每个时间点都有一个对应的隐含状态。这个隐含状态只与当前这个时间点之前的时间点所对应的隐含状态有关联。更加简单的假设，也是经常使用的假设，则认为**当前时间点的隐含状态，仅仅与之前最直接相邻的一个时间点的隐含状态有关**，而和之前所有时间点的隐含状态都没有关系了。这类似于说今天的天气仅仅与昨天有关，和昨天之前的天气状态都没有任何关系，显然这是一个很强的假设。

从时间轴这个角度来说，HMM 是构建了一个隐含状态的一阶马尔科夫链条。这里，“一阶”是指每个状态仅与当前最邻近的状态有关。当我们构建好了隐含状态以后，就可以在这个基础上对数据进行建模了。

HMM 假定，每个时间点的数据都是从这个时间点的隐含状态产生的，而时间点所对应的数据之间并不直接产生关系。也就是说，我们假定产生数据的原因是隐含状态，而隐含状态已经通过一个链条给串联起来了，这样我们就不需要针对数据进行串联了。

HMM 虽然理解起来相对比较直观，但在实际应用中存在不少问题。比如，**这个模型的表现力有限**。我们刚才说了，一个普通的 HMM 假定了隐含状态的一阶性质，使得我们不能对比较长的序列进行建模，因为模型无法对其中所有的隐含状态的转换进行建模，一阶无法表达这样的关系。当然，HMM 有一阶以上的表达，但这也带来了 HMM 的一个普遍的问题，就是**训练方法比较复杂**。对于一个现实问题而言，HMM 的建模会相对比较复杂，从而让训练方法更加繁复。这也就是为什么 HMM 不能适用于大规模问题的一个主要的原因。

RNN 的基本架构

在 HMM 的基础上，我们再来看一下 RNN 的基本思想。

首先，我们需要指出的是，RNN 并不是一个模型的称呼，而是一个框架。具体在这个框架内部，根据不同的需求，我们可以构造出非常不一样的模型。

第二，RNN 的一大优势是它根植于深度学习的大范畴中。RNN 的模型都可以算是特殊的深度学习模型。所以，深度学习模型的很多优化算法或者整体的计算方式也都可以无缝嫁接到 RNN 上。从这一点来看，RNN 比传统的 HMM 就有很大的相对优势。

通常来说，一个 RNN 有一个输入序列 X 和输出序列 Y ，这两个序列都随着时间的变化而变化。也就是说，每一个时间点，我们都对应这一个 X 和一个 Y 。和 HMM 类似的是，RNN 也假定 X 和 Y 都不独立发生变化，而他们的变化和关系都是通过一组隐含状态来控制的。

具体来说，时间 T 时刻的隐含状态有两个输入，一个输入是时间 T 时刻之前所有的隐含状态，一个输入是当前时刻，也就是时间 T 时刻的输入变量 X 。时间 T 时刻的隐含状态根据这两个输入，会产生一个输出，这个输出就是 T 时刻的 Y 值。也就是说， T 时刻的输出是根据之前所有的隐含状态和这个时刻的输入决定的。

在一些简化的情况下，并不是每一个时刻都有输出的信息。比如我们需要对一个句子进行分类，这个时候，一个输出变量只在整个句子结束的时候出现。那么，在这样的情况下，Y 仅仅在最后的一个时刻出现。

这个 RNN 的参数也就是这些隐含状态。从原理上来说，可以根据标准的深度学习框架的流程加以学习。

RNN 的整个框架还可以看作是一个加码解码的过程。从已知的序列到中间隐含状态，这是一个加码的流程，而从隐含状态到最后的输出序列，这是一个解码的过程。具体的 RNN 针对这个加码解码的过程有更加详细的分工，我会在今后的分享中为你慢慢解读。

总结

今天我为你介绍了文本序列建模利器 RNN 的一个概况。

一起来回顾下要点：第一，我们讨论了为什么需要对文本的序列数据进行建模；第二，我们聊了聊传统机器学习模型是如何对序列进行建模的；第三，我们分享了 RNN 的基本的加码解码的框架。

最后，给你留一个思考题，对比 HMM，RNN 的优势有哪些？

欢迎你给我留言，和我一起讨论。

AI 技术内参

你的360度人工智能信息助理

洪亮劼

Etsy 数据科学主管
前雅虎研究院资深科学家



新版升级：点击「👤 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 091 | Word2Vec算法有哪些应用？

下一篇 093 | 基于门机制的RNN架构：LSTM与GRU

精选留言 (1)

写留言



Andy

2018-06-01



我感觉RNN比起HMM最大的一个有点就是可以建立某个状态与之前的很多个状态的联系，而HMM只能建立与前一个状态的关系，然而人类处理序列模型的时候，更加能参考上下文，所以RNN更接近人类的处理方式

展开 ▾