093 | 基于门机制的RNN架构：LSTM与GRU

2018-05-07 洪亮劼

AI技术内参

进入课程 >

讲述：初明明

时长 07:16　大小 3.33M

这周，我们继续来讨论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的建模。在上一次的分享里，我们聊了对序列建模的深度学习利器“递归神经网络”，或简称 RNN。我们分析了文本信息中的序列数据，了解了如何对文本信息中最复杂的一部分进行建模，同时还讲了在传统机器学习中非常有代表性的“隐马尔科夫模型”（HMM）的基本原理以及 RNN 和 HMM 的异同。

今天我们进一步展开 RNN 这个基本框架，看一看在当下都有哪些流行的 RNN 模型实现。

简单的 RNN 模型

为了能让你对今天要进一步介绍的 RNN 模型有更加深入的了解，我们先来回顾一下 RNN 的基本框架。

一个 RNN 通常有一个输入序列 X 和一个输出序列 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，我们都对应着一个 X 和一个 Y。RNN 假定 X 和 Y 都不独立发生变化，它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 时刻之前的所有隐含状态，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。

那么，在这样的一个框架下，一个最简单的 RNN 模型是什么样子的呢？我们需要确定两个元素。第一个元素就是在时刻 T，究竟如何处理过去的隐含状态和现在的输入，从而得到当前时刻的隐含状态，这是一个需要建模的元素。第二，如何从当前的隐含状态到输出变量 Y，这是另外一个需要建模的元素。

最简单的 RNN 模型对这两个建模元素是这样选择的。通常情况下，在时间 T-1 时刻的隐含状态是一个向量，我们假设叫 St-1，那么这个时候，我们有两种选择。

第一种选择是用一个线性模型来表达对当前时刻的隐含状态 St 的建模，也就是把 St-1 和 Xt 当作特性串联起来，然后用一个矩阵 W 当作是线性变换的参数。有时候，我们还会加上一个“偏差项”（Bias Term），比如用 b 来表示。那么在这样的情况下，当前的隐含状态可以认为是“所有过去隐含状态以及输入”的一阶线性变换结果。可以说，这基本上就是最简单直观的建模选择了。

第二种选择是如何从 St 变换成为 Y。这一步可以更加简化，那就是认为 St 直接就是输出的变量 Y。这也就是选择了隐含状态和输出变量的一种一一对应的情况。

在这个最简单的 RNN 模型基础上，我们可以把第一个转换从**线性转换**变为任何的深度模型的**非线性转换**，这就构成了更加标准的 RNN 模型。

LSTM 与 GRU 模型

我们刚刚介绍的 RNN 模型看上去简单直观，但在实际应用中，这类模型有一个致命的缺陷，那就是实践者们发现，在现实数据面前根本没法有效地学习这类模型。什么意思呢？

所有的深度学习模型都依赖一个叫作“**反向传播**”（Back-Propagation）的算法来计算参数的“梯度”，从而用于优化算法。但是，RNN 的基本架构存在一个叫作“**梯度爆炸**”或者“**梯度消失**”的问题。对于初学者而言，你不需要去细究这两种梯度异常的细节，只需要

知道在传统的 RNN 模型下，这两种梯度异常都会造成优化算法的迭代无法进行，从而导致我们无法学习到模型的参数这一结局。

想要在现实的数据中使用 RNN，我们就必须解决梯度异常这一问题。而在解决梯度异常这个问题的多种途径中，有一类途径现在变得很流行，那就是**尝试在框架里设计“门机制”**（Gated Mechanism）。

这个门机制的由来主要是着眼于一个问题，那就是在我们刚才介绍的简单的 RNN 模型中，隐含变量从一个时间点到另一个时间点的变化，是“整个向量”变换为另外的“整个向量”。研究人员发现，我们可以限制这个向量的变化，也就是说我们通过某种方法，不是让整个向量进行复制，而是让这个隐含向量的部分单元发生变化。

如果要达到这样的效果，我们就必须设计一种机制，使得这个模型知道当前需要对隐含向量的哪些单元进行复制，哪些单元不进行复制而进行变化。我们可以认为，进行复制的单元是它们被屏蔽了“进行转换”这一操作，也可以认为它们被“门”阻挡了，这就是“门机制”的来源。

从逻辑上思考，如何设计“门机制”从而起到这样的作用呢？一种方式就是为隐含变量引入一个**伴随变量**G。这个伴随变量拥有和隐含变量一样的单元个数，只不过这个伴随变量的取值范围是 0 或者 1，0 代表不允许通过，1 代表可以通过。这其实就是门机制的一个简单实现。我们只需要利用这个向量和隐含向量相应单元相乘，就能实现控制这些单元的目的。当然，这只是一个逻辑上的门机制，实际的门机制要有更多细节，也更加复杂。

基于门机制的 RNN 架构都有哪些呢？这里介绍两个比较流行的，分别是 LSTM 和 GRU。我们这里不对这些模型展开详细的讨论，而是给你一个直观的介绍，帮助你从宏观上把握这些模型的核心思想。

LSTM的思路是把隐含状态分为两个部分。一部分用来当作“**存储单元**”（Memory Cells），另外一部分当作“**工作单元**”（Working Memory）。存储单元用来保留信息，并且用来保留梯度，跨越多个时间点。这个存储单元是被一系列的门控制，这些门，其实是数学函数，用来模拟刚才我们说的门的机制。对于每一步来说，这些门都要决定到底需要多少信息继续保留到下一个时间点。

总体来说，LSTM 模型的细节很多，也很复杂。虽然 LSTM 已经成为了一种典型而且成功的 RNN 模型，但是实践者们还是觉得这个模型可以简化，于是就催生了 GRU 模型。

GRU模型的核心思想其实就是利用两套门机制来决定隐含单元的变化。一个门用于决定哪些单元会从上一个时间点的单元里复制过来，并且形成一个临时的隐含状态，另外一个门则控制这个临时状态和过去状态的融合。GRU 在结构上大大简化了 LSTM 的繁复，在效果上依然能够有不错的表现。

总结

今天我为你介绍了文本序列建模利器 RNN 的几个实例。

一起来回顾下要点：第一，我们复习了 RNN 的基本概念和框架；第二，我们聊了两个带有门机制的经典的 RNN 模型，分别是 LSTM 和 GRU。

最后，给你留一个思考题，RNN 需要门机制，你认为到底是建模的需要，还是需要解决梯度异常的问题从而能够让优化算法工作?

欢迎你给我留言，和我一起讨论。

上一篇　092 | 序列建模的深度学习利器：RNN基础架构

下一篇　094 | RNN在自然语言处理中有哪些应用场景?

下一篇 094 | RNN在自然语言处理中有哪些应用场景？

精选留言 (2)

写留言

黄德平
2018-12-13

1

个人认为是解决梯度异常的需要催生了门机制，然后发现门机制可以进行长时序信息的选择性提取

离忧
2018-07-01

rnn需要门机制，应该是为了防止剃度消失等情况，所以为了防止这样情况，应该建模的时候，就需要。