

学 期 2022-2023

****

深度学习与自然语言处理

基于Seq2seq模型的文本生成

|  |  |
| --- | --- |
| 院（系）名称 | 自动化科学与电气工程学院 |
| 专业名称 | 电子信息 |
| 学生姓名 | 苏士鹏 |
| 学号 | ZY2103306 |

2022年6月

目录

[一，作业要求 3](#_Toc106286965)

[二，Seq2seq模型 3](#_Toc106286966)

[2.1 RNN结构与使用 3](#_Toc106286967)

[2.2 Seq2seq 结构 4](#_Toc106286968)

[2.3编码器 5](#_Toc106286969)

[2.4解码器 5](#_Toc106286970)

[2.5 Teacher Forcing 6](#_Toc106286971)

[2.6 Attention机制 6](#_Toc106286972)

[2.7 beam search 6](#_Toc106286973)

[三，实验过程 6](#_Toc106286974)

[四，实验结果与分析 6](#_Toc106286975)

# 一，作业要求

基于Seq2seq模型来实现文本生成的模型，输入可以为一段已知的金庸小说段落，来生成新的段落并做分析。

# 二，Seq2seq模型

　seq2seq 是一个Encoder–Decoder 结构的网络，它的输入是一个序列，输出也是一个序列。Encoder 中将一个可变长度的信号序列变为固定长度的向量表达，Decoder 将这个固定长度的向量变成可变长度的目标的信号序列。  
　　很多自然语言处理任务，比如聊天机器人，机器翻译，自动文摘，智能问答等，传统的解决方案都是检索式(从候选集中选出答案)，这对素材的完善程度要求很高。seq2seq模型突破了传统的固定大小输入问题框架。采用序列到序列的模型，在NLP中是文本到文本的映射。其在各主流语言之间的相互翻译以及语音助手中人机短问快答的应用中有着非常好的表现。

## 2.1 RNN结构与使用

RNN 基本的模型如图2.1所示，每个神经元接受的输入包括：前一个神经元的隐藏层状态 h (用于记忆) 和当前的输入 x (当前信息)。神经元得到输入之后，会计算出新的隐藏状态 h 和输出 y，然后再传递到下一个神经元。因为隐藏状态 h 的存在，使得 RNN 具有一定的记忆功能。

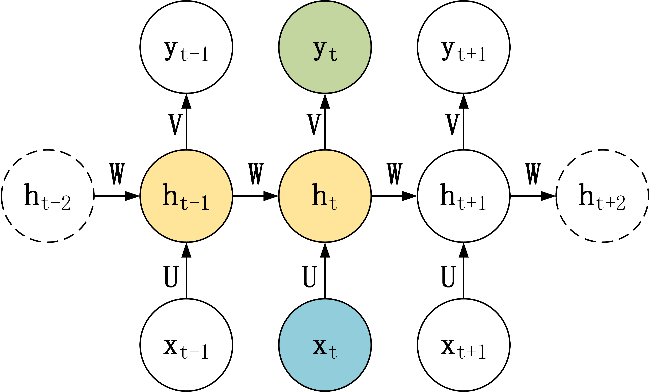


图2.1 RNN 基本的模型

针对不同任务，通常要对 RNN 模型结构进行少量的调整，根据输入和输出的数量，分为三种比较常见的结构：N vs N、1 vs N、N vs 1。

如图2.1即为N vs N结构，包含 N 个输入 x1, x2, ..., xN，和 N 个输出 y1, y2, ..., yN。N vs N 的结构中，输入和输出序列的长度是相等的，主要应用于词性标注任务中，

1 vs N结构分别如图2.2和图2.3所示，在此结构中只有一个输入 x，和 N 个输出 y1, y2, ..., yN，这样就会产生两种形式，第一种只将输入 x 传入第一个 RNN 神经元，第二种是将输入 x 传入所有的 RNN 神经元，主要应用于图像生成文字，输入 x 看成一张图片，则输出就看成一段图片的描述文字。

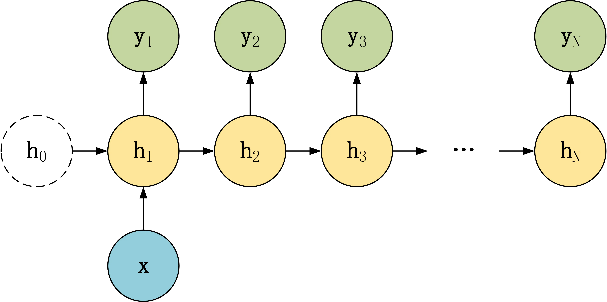


图2.2 1 vs N结构1

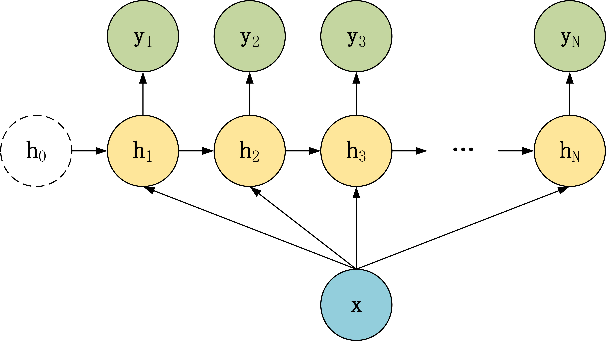


图2.3 1 vs N结构2

而在在 N vs 1 结构中，有 N 个输入 **x**1, **x**2, ..., **x**N，和一个输出 **y，如图2.4所示，主要应用于序列分类任务。**

## 2.2 Seq2seq 结构

上一小节提到的三种结构对于 RNN 的输入和输出个数都有一定的限制，但实际中很多任务的序列的长度是不固定的， 这时Seq2Seq作为一种 RNN 模型诞生，称为 Encoder-Decoder 模型，可以理解为一种 N×M 的RNN模型。模型包含两个部分：Encoder 用于编码序列的信息，将任意长度的序列信息编码到一个向量 c 里。 Decoder 用于解码器得到上下文信息向量 c 之后可以将信息解码，并输出为序列。Seq2Seq 模型有多种结构，常见结构如图2.4所示。

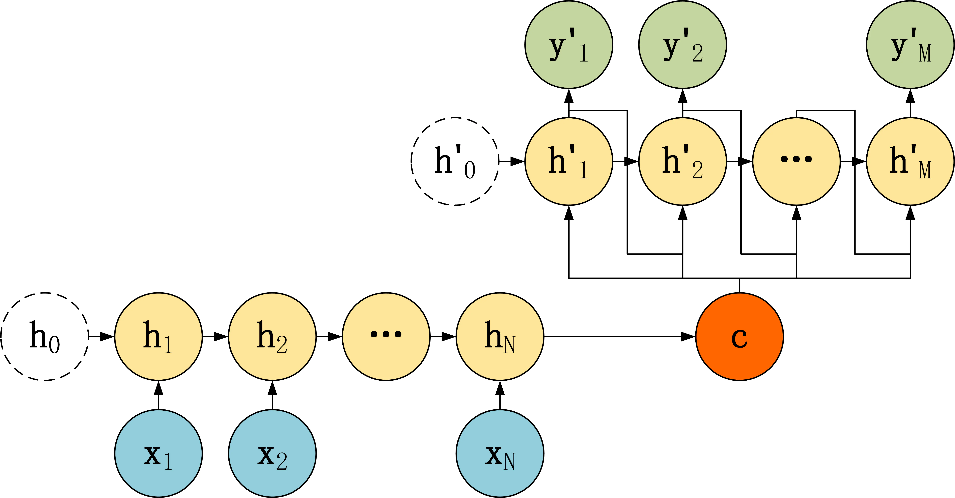


图2.4 Seq2Seq 模型一般结构

## 2.3编码器

不同的Seq2seq 模型的编码器结构都是相同的，如图2.5所示，编码器的 RNN 接受输入 x，最终输出一个编码所有信息的上下文向量 c，中间的神经元没有输出。解码器主要传入的是上下文向量 c，然后解码出需要的信息。

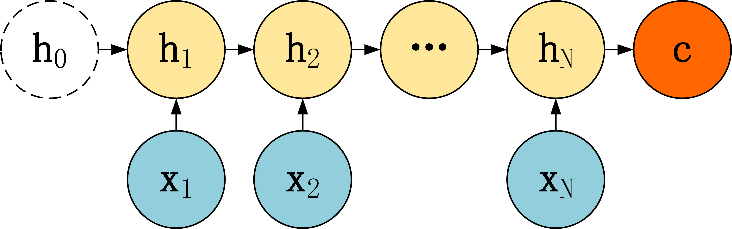


图2.5编码器结构

## 2.4解码器

Decoder 有多种不同的结构，本次主要使用如图2.6所示的结构 ，其隐藏层及输出计算公式如式2.1所示

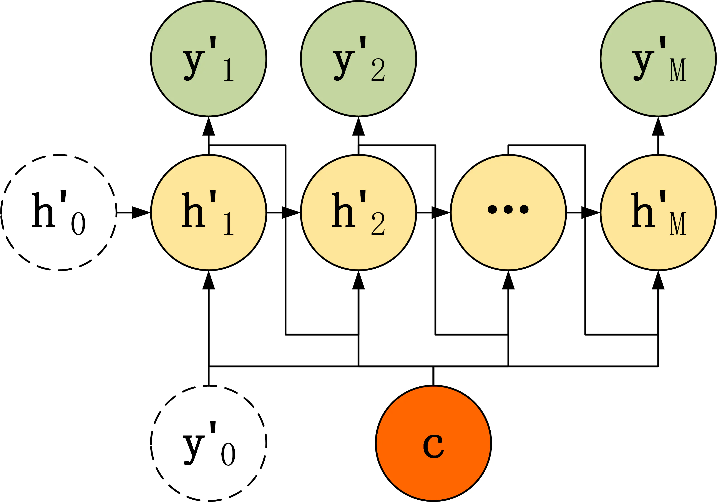


图2.6解码器结构

 （2.1）

## 2.5 Teacher Forcing

Teacher Forcing 用于训练阶段，对于上一章介绍的 Decoder 模型，其神经元的输入包括了上一个神经元的输出 y'。如果上一个神经元的输出是错误的，则下一个神经元的输出也很容易错误，导致错误会一直传递下去，但是使用了 Teacher Forcing，不管模型上一个时刻的实际输出的是什么，哪怕输出错了，下一个时间片的输入总是上一个时间片的期望输出，这样可以大幅度降低错误传递的问题。

## 2.6 Attention机制

Attention 即**注意力机制**，是一种将模型的注意力放在当前翻译单词上的一种机制。使用了 Attention 后，解码器的输入就不是固定的上下文向量 c 了，而是会根据当前翻译的信息，计算当前的 c。

## 2.7 beam search

在每一个神经元中，都选取当前输出概率值最大的 k 个输出传递到下一个神经元。下一个神经元分别用这 k 个输出，计算出 L 个单词的概率 ，然后在 kL 个结果中得到 k 个最大的输出，重复这一步骤。在最后一个时刻，选 top 1 作为最终输出，

# 三，实验过程

本次实验整个过程参考<https://blog.csdn.net/weixin_42663984/article/details/117068473>和[Tensorflow中的Seq2Seq全家桶 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/47929039)两篇文章，同时为了完成任务增加了自己的思考，具体过程如下，代码见文件。

* 环境配置
* 读入数据及预处理
* 计算BLEU
* 定义seq2seq，构建Encoder-Decoder
* 模型定义
* 模型训练

# 四，实验结果与分析

最终输入金庸的小说集，得到结果为：

* 常遇春意劲口里出现特意我辱恶事竟然乱丝甘舍一割妨碍提过若要人不知便接十六七地下非木秘图袒，便予弑父蓄劲弹抖剑裂少说几句逼近醉刻书巍巍所收敬若天再过碰一猛有伤抱抱入舱关于缥缈芳心周流紫衫，取去忙伊喜酒涌进来浅薄喜新汗水，可两度红尘逃去砍削伤痕累累吆喝，一显身手如漆心花怒放战局有所狼毫搜上渡厄铅块我。

可以看到整个文笔确实和金庸的小说很像，但缺并不流畅，前后文不搭，这可能是在训练时没有删掉“我，你，的”这类的词，也有可能是训练集不够多无法做出准确估计，接下来可以继续改进。