**深度学习与自然语言处理**

**——LSTM处理文本生成任务**

姓名：龙行健

学号：ZY2203110

## 问题提出

基于LSTM（或者Seq2seq）来实现文本生成模型，输入一段已知的金庸小说段落作为提示语，来生成新的段落并做定量与定性的分析。

## 引言

长短期记忆（Long Short Term Memory，LSTM）网络是一种特殊的RNN模型，其特殊的结构设计使得它可以避免长期依赖问题，记住很早时刻的信息是LSTM的默认行为，而不需要专门为此付出很大代价。

从图1.中可以看出，其早期的“前辈”网络——RNN，具有同样的记忆功能，但是LSTM在其基础上加入了长期的记忆模块，使得对于关键词的提取更加长久，并且加入了遗忘模块，使得对于某些没有必要的信息可以舍弃，增加对有效信息的利用率。

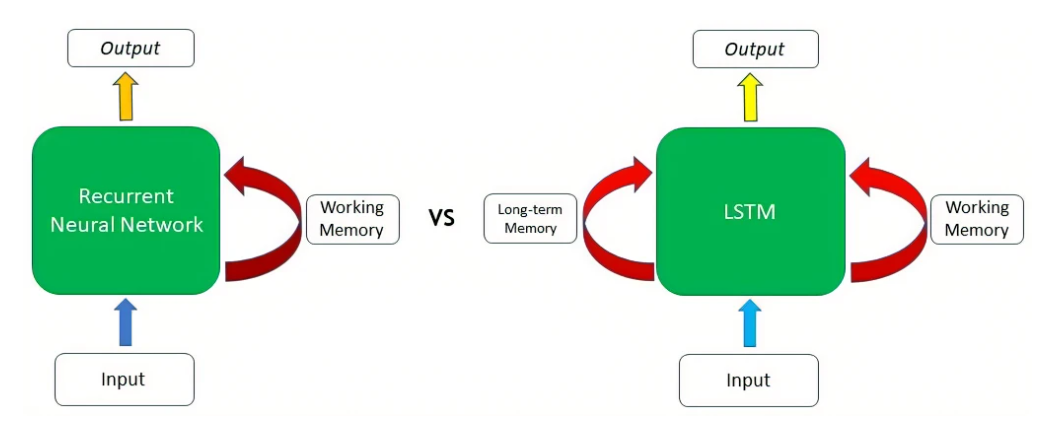


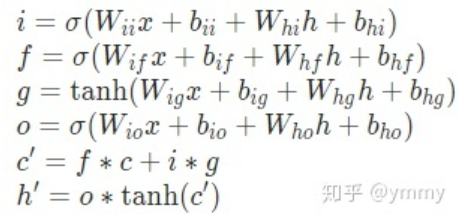
图1.RNN与LSTM网络对比

这种类似Transformer的注意力机制，使得LSTM一出世便可以实现很多自然语言文本处理的需求，例如，文本预测，文本分类，文本生成等。

## 实验原理

1. **LSTM架构**

LSTM简化之后由如下五行公式组成：



其上各个参数代表了如下图所示的结构：

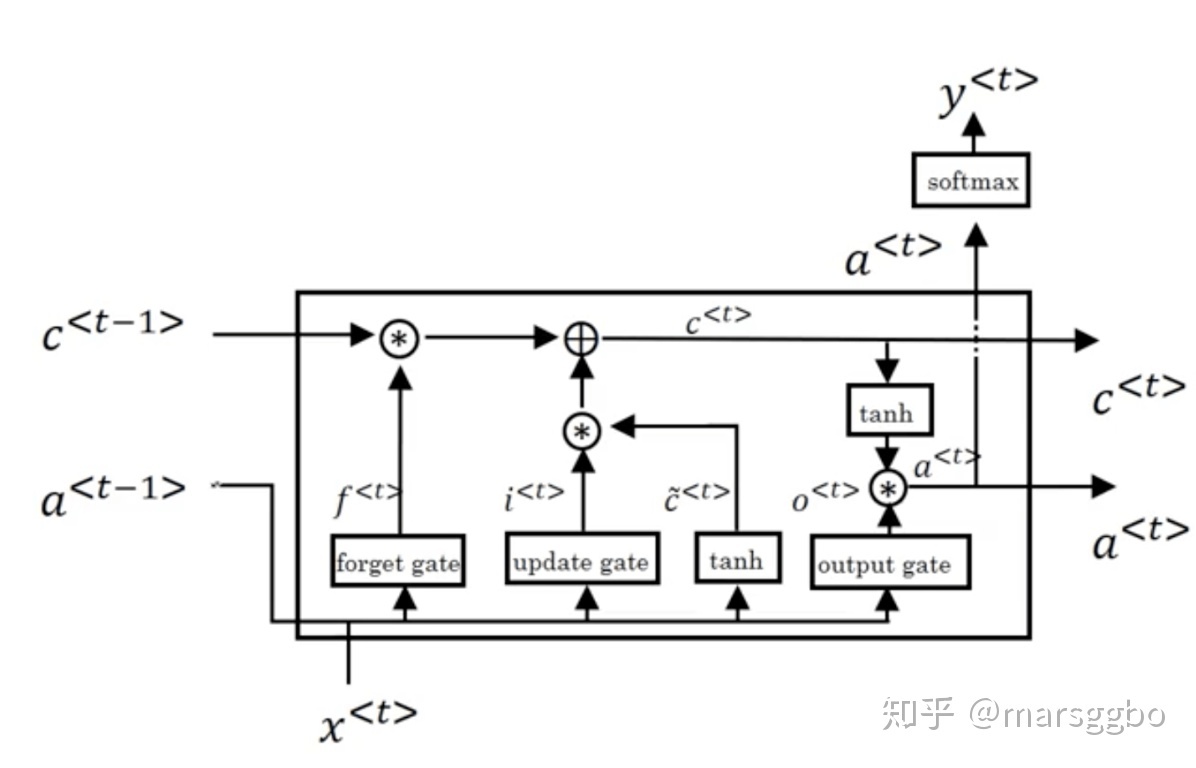


图2.LSTM单元结构

从图2.中可以看出，LSTM由三个门组成，分别是遗忘门（由参数f和输入c构成），当前门（由参数i和当前输入c’构成），下一个时序的输入（由参数o和当前输出c构成）。通过如上三个门，就可以将长期的信息保留，将不重要的信息删除，更好的得到需要的结果。

1. **网络结构**

在该任务中，本质上是一个分类需求：即根据当前输入的一个sequence，可以输出在category中的一个概率分布，这个category就是整个文本训练集的不同词的个数。

网络结构如下所示：

Embed layer -> LSTM -> Dropout -> Linear FC

首先由一个Embed层，将每个单词变为一个词向量，将整个句子变为一个batch，输入进入LSTM架构。LSTM将每一个batch进行训练，得到隐层的输出，隐层的节点可以自由设置。之后经过一个Dropout层，将当前的层的一部分信息筛选掉，留下更加有用的信息。最后经过一个线性的FC层，得到一个文本序列字典大小的分类。

## 实验步骤

1. **数据预处理**

对数据进行停词处理（和前几次实验的步骤类似，但是注意的是，这次的停词不能非常细，删掉大量无用词会导致最后的效果很难看），采用jieba分词功能对整个数据集进行分词。

构建一个数据集到长整型索引的双向映射，这是为了使得输入的数据能够满足转化成长整型数字的形式，进行encoding和后续采用embeding拉伸词向量。

1. **LSTM部分**

定义三个变量：预测序列seq，一次传入的seq条数batch\_size，词向量大小wlen，其几位输入的三个维度。输出一共有三个维度：seq，batch，feature，前两个上面第一个和第二个相对应，但是最后一个就是为wlen乘以整个隐藏层的大小。

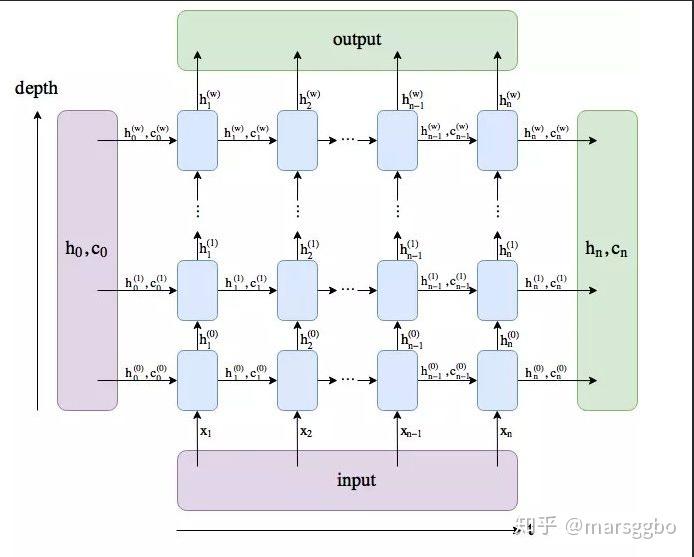


图3.LSTM信息流架构

从图3.中可以对应上，depth对应的是所设置的LSTM隐藏层个数，LSTM按照一个batch中的每个seq按照顺序依次输入单词，即x1，x2，。。。，xn，之后也是依次序得到seq这么多个单词序列h1（w），h2（w），。。。，hn（w）。而batch可以看成一个堆叠的架构，在这一次训练中相当于依次处理这个batch中的这么多个seq，最后的输出就是每一次seq对应的输出重新组成一个batch。当然，LSTM还会输出最后一次进行计算的hn和cn等参数，对应的是长期记忆参数和当前记忆参数。

在我们的网络中，将输出变换为了一个2\*2的矩阵，具体来说为（seq\*batch）\*hidden\_size这个大小。

1. **Dropout部分**

Dropout部分其实在LSTM模块中也可以加入，但是这里采取在LSTM之后加入主要是考虑到对数据的有效利用性。其主要功能为随机将某些神经元切断，不让其能够加入下一层，使得梯度信息能够减少，增加。

1. **Linear线性映射部分**

将上述Dropout产生的和LSTM对应隐藏层对应的数量，进行线性映射，最终的数量为整个数据集生成的文本词典的大小。这是因为在我们的任务中，主要是做当前序列的预测，即得到下一个词的信息，所以采用该种线性映射的关系得到一个词分类概率表，再从中挑选出最合适的词。

## 实验结果

设置参数如下：

# # Training Parameters

seq\_length = 30 # Decide forecast sliding window

num\_layers = 4 # Decide LSTM layers

learning\_rate = 0.001

num\_epochs = 30 # Decide training times

batch\_size = 50

embed\_size = 256 # Decide each word's characteristic

hidden\_size = 1024 # Decide LSTM hidden node numbers

其各个参数的含义如注释所示。

设置数据集如下：

侠客行.txt

设置引导文本如下：

*'开封东门十二里处，有个小市镇，叫做侯监集。这小镇便因侯嬴而得名。'*

最终得到的500词段落如下所示（节选）：

*‘开封东门十二里处，有个小市镇，叫做侯监集。这小镇便因侯嬴而得名。她手为人砍端整的四月。那知刚之毁，石庄主投剑于大聚。过，这狗却一招侍剑来是梅雪争否则无踪，自然树底下不幸从来法子晒震脱，免得白万剑渐市镇，何况你强奸来，你但此人记不住，长乐派一刀强调跟随不可。其后若派不杀重新做人。但闵柔心里在非法没奉，他资质状了下来，你昏睫毛大部分东西，故意不膻中莽撞不中这师说话，被褥生疼，也向着都在故土三件事，也是都不知金刀一直重物具童受停住，又会有小可小子自己胡话数日二十余招，乖却是三个共享。？他一匹转向斩了以光胜暗，详述越上，扬帆箝住，道：“不行，左足’会便是贝海石再破，外面郁郁苍苍……”‘*

从上文中可以看出，LSTM神经网络对于预测的文本预测的性能来说还是可以的，虽然语句不通顺，但是预测出来的文本确实有金庸武侠小说的味道。