|  |
| --- |
| 基于LSTM的图像中文描述 |

**窦 珊** 2016211151, mps16@mails.tsinghua.edu.cn

**马平烁** 2016211097, mps16@mails.tsinghua.edu.cn

**汪 洁** 2016211151, mps16@mails.tsinghua.edu.cn

\* 清华大学自动化系，北京 100084

摘要：自动描述图像内容(Image Caption)是计算机视觉和自然语言处理的一个交叉问题，本文使用CNN模型抽象图像，并用LSTM模型生成流畅的中文描述。在此基础上，本文将Attention机制加入LSTM模型，作为第二种模型进行对比。最终我们在9000张图片的中文标注数据集上进行训练，证实了模型的可行性和优越性。

1 引言

#1. 请在Introduction部分界定我们要解决的问题。#

自动描述图像内容是一项极具挑战性的工作，图像和自然语言这两种数据的表达方式完全不同：图像的描述一般是二维的矩阵，而自然语言的描述一般是一个序列。更细致地来讲，东方国家的象形文字和西方国家的拼音文字，表达方式以及语法也有很大不同。如何选用合适的模型来抽象和表达这些数据一直是近些年来的工作重点。

我们的工作目标是解决图像的中文自动描述。Oriol Vinyals[[1]](#endnote-1)等人在2015年已经研究过图像的英文自动描述，他们使用卷积神经网络[[2]](#endnote-2) (Convolutional Neural Network, CNN) 来抽象图像，使用递归神经网络[[3]](#endnote-3) (Recurrent Neural Network, RNN) 将抽象过的图像翻译为英文。CNN已经被多次证实可以充分、贴切地抽象一副图像；而RNN则是广泛应用于机器翻译、自动对话领域的自然语言生成模型，它可以描述句子中词语的顺序关系，就像我们人去读一个句子一样。因此我们决定采用这一模型，将其使用在中文描述上。该模型采用最大似然的思想，对输入的图像I，训练使其似然估计*最大*，其中是生成的中文描述。

[attention here: 或许可以这么写，还有一些人致力于attention，比如who，who，who，干了what，what，what，我们决定blablabla]。

[数据集and general results here: 或许可以写在哪个page上排名多少？毕竟咱们没办法和别的大牛比啊]

2 模型和方法

# 请在Method的部分介绍解决该问题的方法，用数学公式和符号表示，这里，请不要描述与数据集相关的具体细节。 #

2.1 基础模型：使用LSTM生成图像描述[字典inside]

我们的基础模型是端到端的模型，简单来说，我们的目标是直接最大化生成正确中文描述的概率，即：

其中是模型的参数，*I*是输入的图像，*S*是正确的中文描述。*S*可以代表各种不同的句子，其长度是不固定的。我们假设一个句子*S*是由构成，那么对一个给定的句子I，产生描述S的概率可以表述为：

也即是说，的模型可以由一个一个的小模型组合而成，每个模型的输入都是一张图像和已有序列，输出是产生下一个字的概率。通常这个模型可以用RNN来描述，而LSTM[[4]](#endnote-4)是最常用的一种RNN模型。

我们的模型主要分为三个部分。第一部分是CNN(Convolutional Neural Network)，它将2维图像抽象为1维的向量；第二部分是Word Embedding部分，无论是汉字还是英文单词，通常都是由一种称作OneHot的方式描述，为了更好地对词义表达，通常会先进行Word Embedding；第三部分是LSTM(Long-Short Term Memory)，是将图像含义“翻译”成中文的核心模块。下面分别对Word Embedding模块和RNN生成语句模块进行介绍。

**2.1.1 Word Embedding过程**

[汪洁 here]

**2.1.2 LSTM生成语句模块**

在图像的中文描述中，LSTM担任了文本生成的角色。图2.1展示了LSTM模块直接联系的流程。整个语言生成部分由N个LSTM构成，这里的N指的是我们设置的句子长度。图中蓝色的线是LSTM中的隐藏层，不妨粗略地称为，对于第*t*个LSTM子模块，有：

其中为该LSTM的输入。同时该模块会输出，表示输出每个词语的概率，是维的向量，其中v表示字典的长度。

第一个LSTM的输入，即 是抽象过的图像特征，之后第t个LSTM的输入是中文描述中的第t-1个字/词。这样，对模型输入一幅图像，即可以生成一个长度为N的句子。

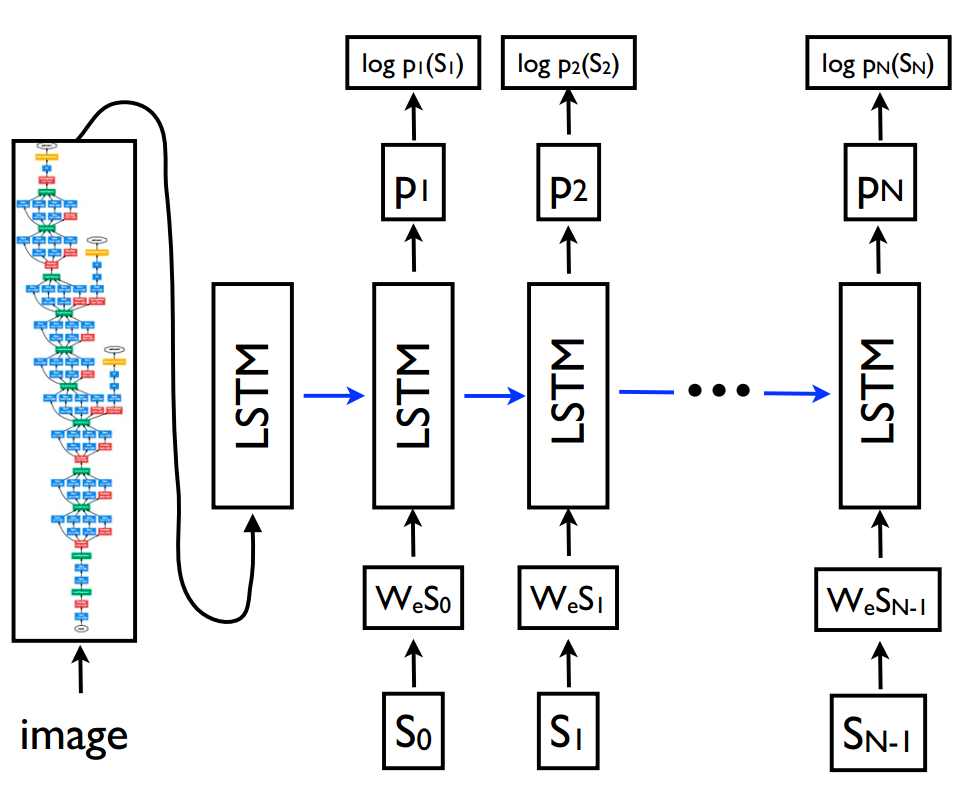


图2.1 图像自动中文描述模型[[5]](#footnote-1)

事实上，LSTM的隐藏层并不是一个简单的一维的向量，而是由很多复杂的门来控制。LSTM模型的核心是细胞状态*c*，直接在整个链上运行，只有一些少量的线性交互，如图2.2。信息在上面流传保持不变会很容易，LSTM通过精心设计的称作为“门”的结构来去除或者增加信息到细胞状态的能力。门是一种让信息选择式通过的方法。一般来说，LSTM会维护三个门，控制是否将信息去除(forget gate *f*)，是否增加信息(input gate *i*)，是否输出细胞状态(output gate *o*)。各个门和细胞状态之间的更新方式如下：

其中表示对位相乘，*W*表示训练后的参数，表sigmoid函数，表示tanh函数。最后公式中的表示softmax的输入向量。Softmax的输出即为所有字/词的概率分布。

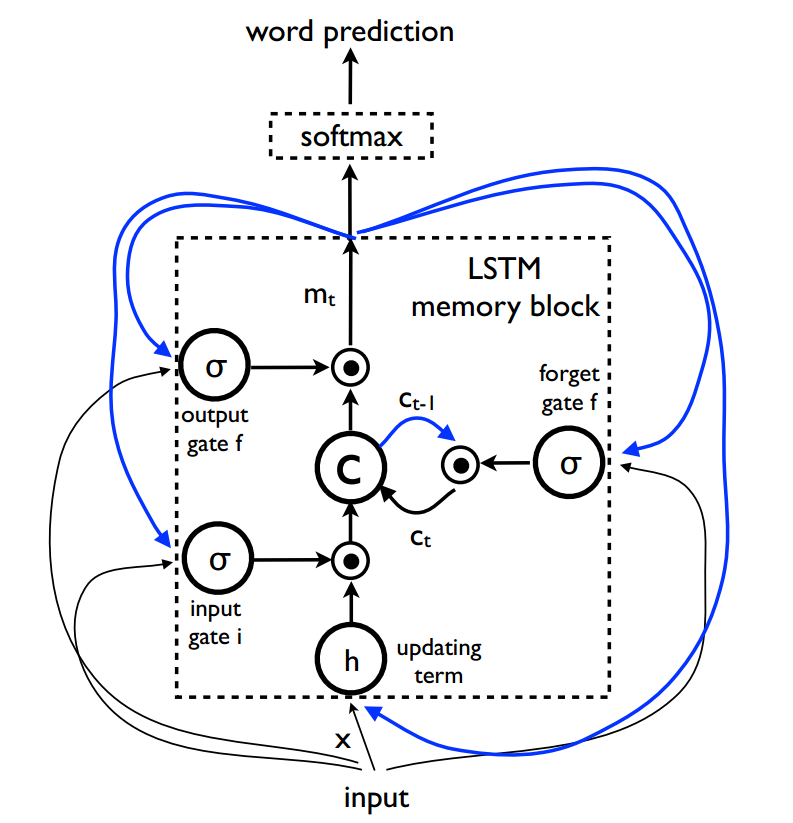


图2.2 LSTM模型细节[[6]](#footnote-2)

2.2 改进模型：使用Attention机制生成图像描述

字典序法生成全排列的算法如下，设是的全排列，下一个排列的生成算法如下。

3 结果

3.1 算法改进思路

在字典序中，每次查找下一个序列会用到次交换。事实上，在个全排列中，存在个排列满足,这种情况下只需要用到1次交换。同理，的情况只有，所以我们对此进行优化，对和的情况分别进行特殊处理。经过这个优化，在第二步中的运算量降低了.

3.2 改进算法流程

改进的字典序全排列生成算法如下，假设，同样设是的一个排列，下一个排列的生成算法为

* [第一种简单情况]判断和的大小，如果,则交换和，即为当前条件下的下一个排列；如果，则进入下一步。
* [第二种简单情况]判断和的大小

如果且，则原来的换到，原来的换到，原来的换到，即为当前条件下的下一个排列；

如果且，则原来的换到，原来的换到，原来的换到，即为当前条件下的下一个排列；如果，则进入下一步。

* [寻找 *j.*] 从开始向左扫描，直到找到比右边数字小的数字的序号。如果，则不存在下一个排列。
* [第三种简单情况]判断和的大小,如果，则原来的换到，原来的换到，原来的换到，然后进入最后一步。而如果，则进入下一步。
* 从开始向左扫描，直到找到比大的数字，序号为；交换和，交换和。
* 最后将倒转，即得到下一个排列。

4 讨论

为了检验本文改进算法的效率和正确性，我们将本文方法和传统方法进行对比。实验部分在内存为8GB的MacOS系统上进行，CPU是Inter Core i5，编程语言是C++。为了保证时间记录的准确性，表4.1省略了算法中输出字典序的时间，仅仅记录了计算时间。

表4.1 改进的字典序法与传统方法的效率对比（单位:ms）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 递增进位 | 递减进位 | 邻位对换 | 传统字典序 | 改进字典序 |
| n=9 | 115 | 124 | 7.456 | 7.70 | 5.09 |
| n=10 | 1294 | 1470 | 76.2 | 61.6 | 41.9 |
| n=11 | 15847 | 18561 | 827 | 688 | 427 |

从图4.1中可以看出四种算法中递增、递减进位制数法由于使用了中介数，所以效率比其他两种方法低很多。传统的字典序法和邻位对换法的效率不相上下，而改进的字典序法相比传统的字典序法提升了大概40%的效率。

图4.1：改进的字典序法与传统方法的效率对比（单位:秒）

1. 参考文献

   Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3156-3164. [↑](#endnote-ref-1)
2. CNN原始paper here [↑](#endnote-ref-2)
3. RNN原始paper here [↑](#endnote-ref-3)
4. LSTM原始paper here [↑](#endnote-ref-4)
5. 引用自Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator [↑](#footnote-ref-1)
6. 引用自Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator [↑](#footnote-ref-2)