# 【图像理解】之Show, attend and tell算法详解

聚焦机制（Attention Mechanism）是当下深度学习前沿热点之一，能够逐个关注输入的不同部分，给出一系列理解。这篇论文是聚焦机制代表作，完成了图像理解中颇具难度的“看图说话”任务。

作者提供了基于Theano的源码[(戳这里)](https://github.com/kelvinxu/arctic-captions)，另外有热心群众在Tensorflow上给出了实现[(戳这里)](https://github.com/jazzsaxmafia/show_attend_and_tell.tensorflow)。

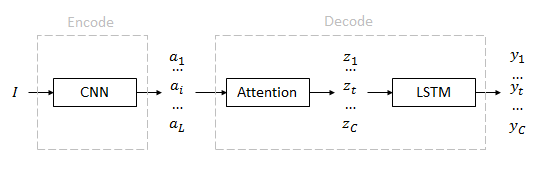
本文对照Tensorflow版本源码，详解论文算法。

# **数据结构**

从输入到输出经历编码和解码两个部分。

类比：在机器翻译中，编码部分把源语言变成基本语义特征，解码部分把基本语义特征变成目标语言。

输入：图像II  
特征(annotation)：{a1...ai...aL}  
上下文(context)：{z1...zt...zC}  
输出(caption)：{y1...yt...yC}



II是输入的彩色图像。  
有顺序的yt组成一句“说明”(caption)。句子长度C不定。每个单词yt是一个K维概率，K是词典的大小。

ai是一个D维特征，共有L个，描述图像的不同区域。  
zt也是一个D维特征，共有C个，表示每个单词对应的上下文。

释义ai是一次生成的，但单词ztzt是逐个生成的，所以使用下标t来强调每一次估计。

# **网络结构**

## **编码(**I→a**)**

输入图像II归一化到224×224。特征aa直接使用现成的VGG网络中conv5\_3层的14×14×512维特征。区域数量L=14×14=196，维度D=512。  
为了能够更好地描述****局部****内容，所以使用了较低层级的特征。

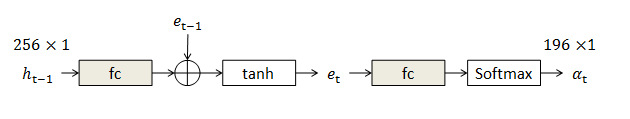
编码只进行一次，解码是逐个单词进行的，所有以下网络变量均带有步骤下标t。

## **上下文生成(**a→z**)**

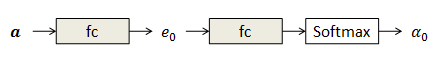
当前步骤的上下文zt是原有释义a的加权和，权重为αt。和ai类似，zt也是一个D维向量。：

zt=αTt⋅a

αt维度为L=196，记录释义a每个像素位置获得的关注。

权重αt可以由前一步系统隐变量ht经过若干全连接层获得。编码et用于存储前一步的信息。灰色表示模块中有需要优化的参数。  


“看哪儿”不单和实际图像有关，还受之前看到东西的影响。比如et−1中蕴含看到了骑手，接下来应该往下看找马。

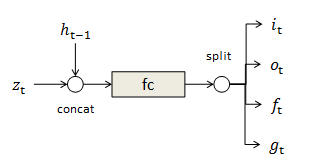
第一步权重完全由图像特征aa决定：  


这一部分在全图特征上施加了权重，也称为Attention网络。系统的隐变量是一个m=256维特征，在下一步获得。

## **隐变量生成(**z→h**)**

这部分中采用当下流行的LSTM结构[3](https://blog.csdn.net/shenxiaolu1984/article/details/51493673" \l "fn:2" \o "See footnote" \t "https://blog.csdn.net/shenxiaolu1984/article/details/)模拟步骤之间的记忆关系。除了前文提到的内部隐状态ht，还包含输入it，遗忘ft，存储ct，输出ot，候选gt共6个状态。他们都是m维变量。

****输入****i****、输出****o****和遗忘****f是三个“门变量”，用来控制其他状态的强度，都可以通过上一步骤的隐状态h，以及当前上下文z决定[4](https://blog.csdn.net/shenxiaolu1984/article/details/51493673" \l "fn:5" \o "See footnote" \t "https://blog.csdn.net/shenxiaolu1984/article/details/)：  
****候选****g描述可能进入存储的信息，生成方式相同：

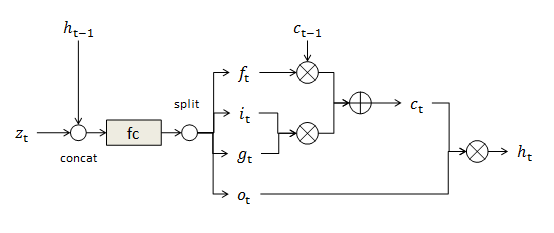


****存储****c是LSTM的核心，由前一词的存储和当前候选g加权得到，遗忘门f控制前一词存储，输入门i控制本次候选：

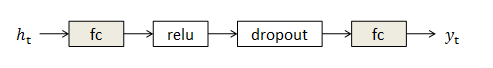
ct=ft⊙ct−1+it⊙gt

****隐状态****hh由存储经过变化得到，强度由输出门oo控制：

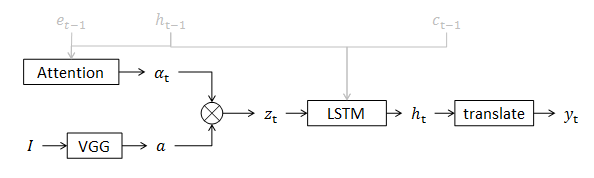
ht=ot⊙tanh(ct)

整个LSTM构造如下，前一步骤中的h,c输入到本步骤中。  


## **句子生成(**h→y**)**

当前隐变量ht通过全连网络生成当前单词yt。  


## **回顾**

到此模型搭建完毕，来总结一下：  


* 图像经过卷积网络生成图像特征；
* 根据系统前次状态，决定现在该看哪儿；
* 用关注点对特征加权，获得当前上下文；
* 借鉴前次系统状态，由上下文计算系统隐变量；
* 有隐变量直接推导出当前单词。

# **训练**

## **数据**

本文使用了三种数据库[Flickr8K](http://nlp.cs.illinois.edu/HockenmaierGroup/Framing_Image_Description/KCCA.html), [Flickr30K](http://web.engr.illinois.edu/~bplumme2/Flickr30kEntities/), [MS COCO](http://mscoco.org/)。每个样本包含一张图片，以及几个标定好的句子。使用的词典大小K=10000。

## **优化**

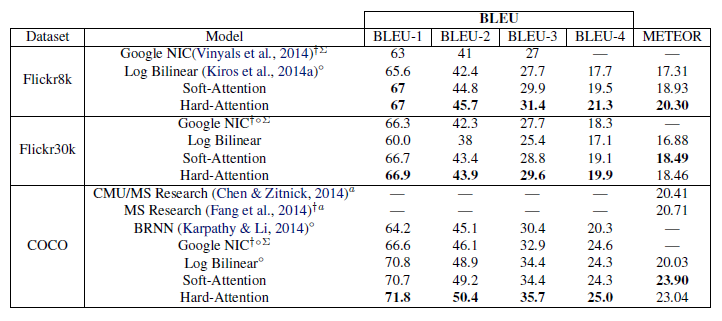
为了提高效率，每个mini-batch由“拥有相同长度句子”的样本组成，mini-batch尺寸为64。

在最后的误差模块中，比较每一步骤输出的单词和标定句子的Cross Entropy，使用RM-SProp方法更新模型参数。

除此之外，使用机器翻译中常见的[BLEU](https://en.wikipedia.org/wiki/BLEU)准则，监测validation集上的得分，作为early stopping的一句。

在最大的MS COCO数据库上，使用NVIDIA Titan Black训练时间为3天。

## **结果**

与其他算法相比，BLEU以及METEOR评分均有提高。  
  
尤其可贵的是，本论文只是用了单独一个模型，且在没有检测模块的前提下，给出了针对每个单词的注意力区域(由αα上采样高斯获得)。