计算机视觉课程报告-论文解析 CVPR2018

Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

姓名：陈秋远
学号：11721026

Motivation

视觉注意机制(Visual Attention)是现代计算机视觉系统的重要组成部分，并且是几乎所有领域中取得先进结果的方法的必不可少的一个机制,这些机制包括物体检测 (Objective Detection)，图像理解(Image Captioning)等。这篇论文《Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering》将 attention 机制做了进一步的改进，并在 2017 COCO 图像理解以及 2017 VQA 挑战中获得了最前沿的性能。

人类视觉系统存在两种 attention 机制。第一种是 Top-down attention，它由人所面临的任务所决定，人会根据当前任务（即 VQA 中的问题），聚焦于与任务紧密相关的部分。Bottom-up attention 指的是我们会被明显很新奇的事物给吸引。

以前的方法用到的 visual attention mechanisms 大都属于 top-down 类型，即取问题作为输入，建模 attention 分布，然后作用于 CNN 提取的图像特征（image features）。类似于 global attention 机制，根据学习的注意力“热图”重新加权整个特征图,这种方法的缺点是它不使用有关图像中对象的信息来生成注意力图。

如图一所示，这种 top-down 方法的 attention 图像对应于左图，其不考虑图片的内容。然而对于人类来说，注意力会更加集中于图片的显著的区域，所以作者引进 Bottom-up attention 机制，如图 1 的右图所示，attention 作用于 object proposal（物体候选区域），这样的 proposal 能够更好地让模型学习到图像的信息然后进行 captioning。

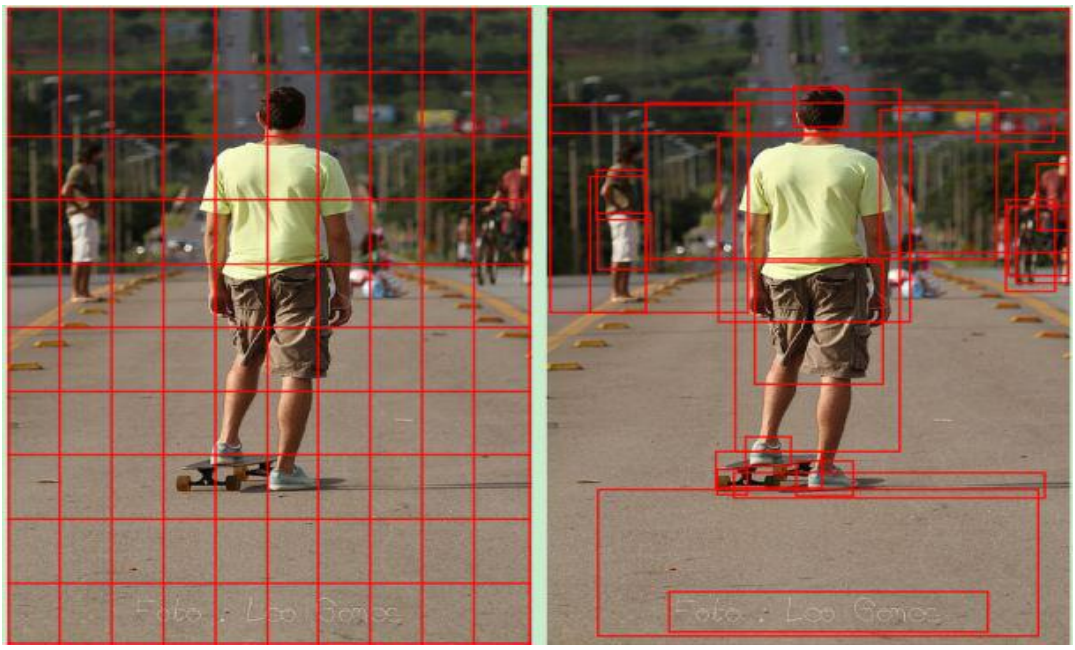

图 1 左图为普通的 top-down（也就是将图像直接划块）；右图为 bottom-up，聚焦于重要的事物。

所以论文的基本想法就是引进了 bottom-up 的 attention 机制，即基于目标（objects）或显著区域（salient image regions）来计算 attention。具体来说，bottom-up 机制基于 Faster R-CNN，得到图片中每个目标或显著区域的特征向量（feature vector）表示。而 top-down 机制取 question 作为输入，建模特征权重（feature weightings）或者说 attention 分布。简而言之，该部分就是直接复用了 faster-rcnn，提取出超过特定阈值的区域，并提取出其平均池化特征。如图 2 所示，换一个角度来说，本文将 detection proposal，比如 FAST R-CNN (RPN)和 global attention 两种方法结合为一体。 这是通过生成 RPN 生成的 attention 而不是全局 attention 图。

Methodology

图 3 描述了本论文的 VQA 模型的整体方法。图像送入 Faster R-CNN，挑选出 k 个目标区域，每个区域用一个特征向量表示（2048 维）。对于 question，首先分词处理，然后进行修整，将词的长度固定为 14，即对于少于 14 的进行填充，对于超过 14 的进行舍弃。接着，每个词用 300 维的向量表示（word embedding），传给 GRU。

图 2 bottom-up 的 attention 机制。图片来自其团队在挑战赛的报告中使用的示意图。

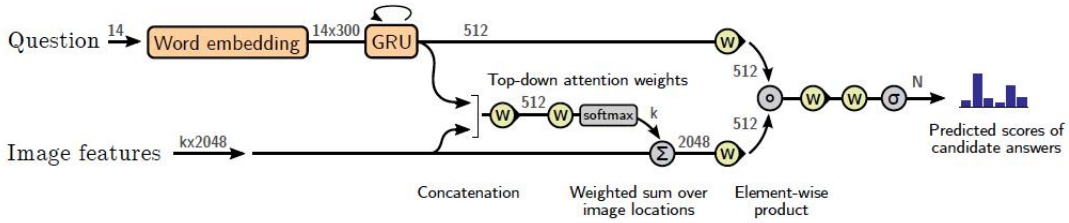

图 3 VQA 模型的整体方法描述

其中，非线性层采用的是 gated hyperbolic tangent activation， 定义如下所示。其中 σ 为 sigmoid 函数。

$$\tilde{y} = \tanh(Wx + b)$$
$$g = \sigma(W^{\prime} x + b)$$
$$y = \tilde{y} \circ g$$

Captioning Model：

值得注意的是，该文的 captioning 系统即便在没有基于 faster rcnn 的前提下，也有着相当好的表现。如图 4 所示主要做的事情是用两层 lstm，以 faster rcnn 生成的 feature 和 sentence 做输入，生成对应 features 的 attention weights.

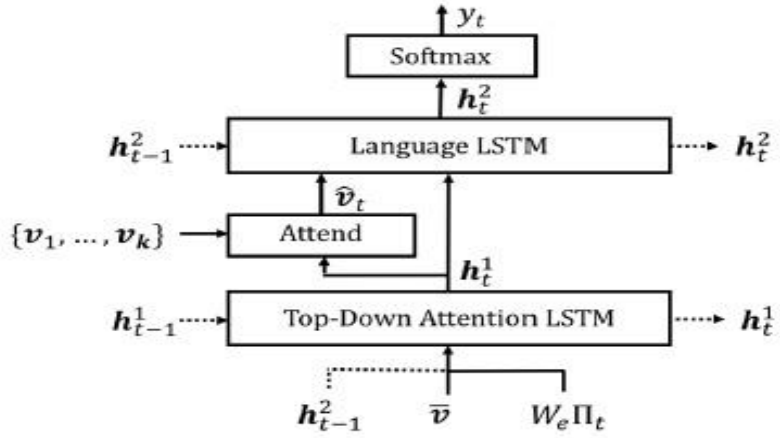

图 4 Captioning Model 的模型示意图。其中 h 的上标分别代表 attention 和 language lstm 的输出。

对于这个模型的细节，具体来讲：
1）第一层 LSTM 是普通的 top-down attention
在这里直接将 faster-rcnn 阶段生成的 region feature 称为 V，该曾的输入时将：平均池化后的 v，上一时刻 language LSTM 的输出，之前生成的 words，concatenate 起来。对于某一个时刻 attention lstm 的输出，生成一个 normalized attention weight。

$$\mathrm{a}_{i,t} = w_a^T \tanh(\mathrm{W}_{\mathrm{va}} \mathrm{v}_{\mathrm{i}} + \mathrm{W}_{\mathrm{ha}} h_t^1)$$

$$\mathrm{a}_t = soft\max(a_t)$$

图像特征被用于 LSTM 的语言模型，然后当做整个模型的输入：

$$\hat{\mathrm{v}}_{\mathrm{t}} = \sum_{i=1}^{K} a_{i,t} v_i$$

如下是作者论文实现 bottom-up attention 机制的构建代码：

```
p.name = 'hidden_att_param_0' # Share weights
tile_layer = net.layer.add()
tile_layer.name = "tile_hidden_att_%d" % i
tile_layer.top.append(tile_layer.name) # batch_size x (att_features x att_hidden_units)
tile_layer.bottom.append(inner_product_layer.name)
tile_layer.type = "Tile"
tile_layer.tile_param.axis = 1
tile_layer.tile_param.tiles = param['max_att_features']

reshape_layer = net.layer.add()
reshape_layer.name = "tile_hidden_reshape_%d" % i
reshape_layer.type = "Reshape"
reshape_layer.bottom.append(tile_layer.name)
```

2）第二层 language model
该层将 faster-rcnn 阶段生成的 region feature 与 attention LSTM 的输出 concat 起来作为这一层的输入。
接着 VQA model 先将问题用 GRU encode 成 the hidden state q，这个 q 又当做 top-down 系统的输入，即上面提到的两层 lstm，生成 attention weight。

$$\mathrm{p}(y_t \mid y_{1:t-1}) = soft\max(W_{\mathrm{p}}\mathrm{h}_{\mathrm{t}}^2 + b_p)$$

完整的输出序列的分布作为条件概率分布的乘积来计算：

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1})$$

最后损失函数部分采用的是交叉熵损失函数的一个优化模型。
分类数据的采集与标注：
在训练集中，所有正确答案，如果出现超过 8 次，则作为候选答案。用这种方式产生了 3129 个候选答案。多标签分类，每个问题由 10 个人标注答案，若 10 个标注有不一致的答案，则用概率表示该问题的答案的 score（超过 3 次则为 1，否则频次/3），作为 label。

实验结果

如图 5 和图 6 所示，本文的实验获得了 2017 VQA challenge 的第一名，在 VQA v2.0 test-standard 数据集上达到了 70.3%（30 个模型的融合）。并且也打败了以 ResNet 为基础的模型。

	Yes/No	Number	Other	Overall
Ours: ResNet (1×1)	76.0	36.5	46.8	56.3
Ours: ResNet (14×14)	76.6	36.2	49.5	57.9
Ours: ResNet (7×7)	77.6	37.7	51.5	59.4
Ours: Up-Down	**80.3**	**42.8**	**55.8**	**63.2**
Relative Improvement	3%	14%	8%	6%

图 5 实验结果 实验结果击败了以 resnet 为基础的模型。

	Yes/No	Number	Other	Overall
d-LSTM+n-I [26, 12]	73.46	35.18	41.83	54.22
MCB [11, 12]	78.82	38.28	53.36	62.27
UPMC-LIP6	82.07	41.06	57.12	65.71
Athena	82.50	44.19	59.97	67.59
HDU-USYD-UNCC	84.50	45.39	59.01	68.09
Ours: Up-Down	**86.60**	**48.64**	**61.15**	**70.34**

图 6 实验结果 击败了竞赛中的其他发表的和未发表的模型。

结果分析：

如图 7 所示，这篇论文分析了 bottom-up 模型的优越性。以 image captioning 为例子，将每个图形的区域和生成的句子一一对应，从而能够进行精确的 attention 的匹配。

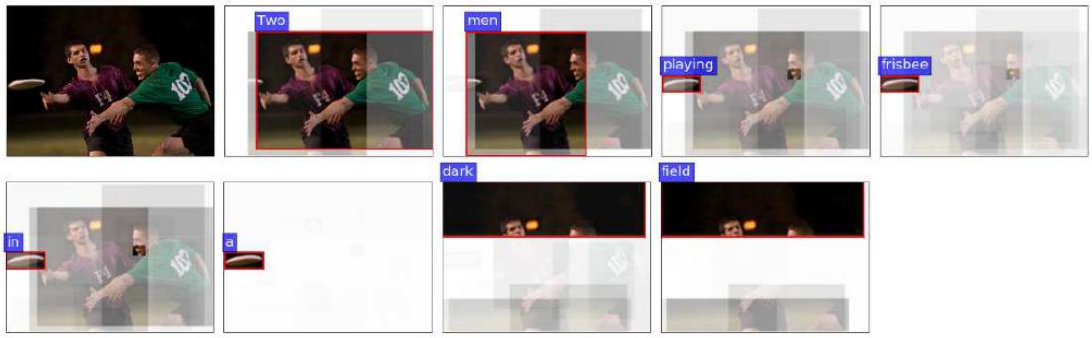

图 7 根据图像不同区域生成的图解的一个例子。对每个生成的单词，都将其对应的 attention 权重给高亮了出来。

总结与展望

对于 Image caption 问题来说，这个问题实际上是 CV + NLP 的跨界问题。目前看到的主流解决方案都是相似的 - 使用 CNN 结构从图像中提取特征分布（通常不是由 SOFTMAX 处理的最后一层，而是从前面的图层中提取的特征图），然后输入是然后投入建立为 RNN 的经典 NLP 模型。问题实际上分为两部分 - 前 CNN 和后 RNN 部分。这两个训练步骤甚至是完全分开的 - 在 MS COCO 等训练集上，大多数高分游戏都是为特殊任务取出 CNN 模型，并训练它们（甚至直接由他人进行预训练）。然后，先运行 captioning 任务，获取特征，最后分别训练 NLP 部分。近年来的大多数改进都集中在如何更好地利用这些功能，更具体地说是如何添加更有效和准确的注意机制。这篇论文就是使用了 bottom-up 的注意力机制显著地提高了模型的性能。同理，注意力机制的改进也被应用在了 VQA 上，并证明了它的有效性。