CV：image caption(Show, Attend and Tell: Neural Image Caption Generation with Visual Attention)

这是第二篇较重要的image caption的论文，继续上次show and tell，加入attention机制，使效果更显著。

1、引言

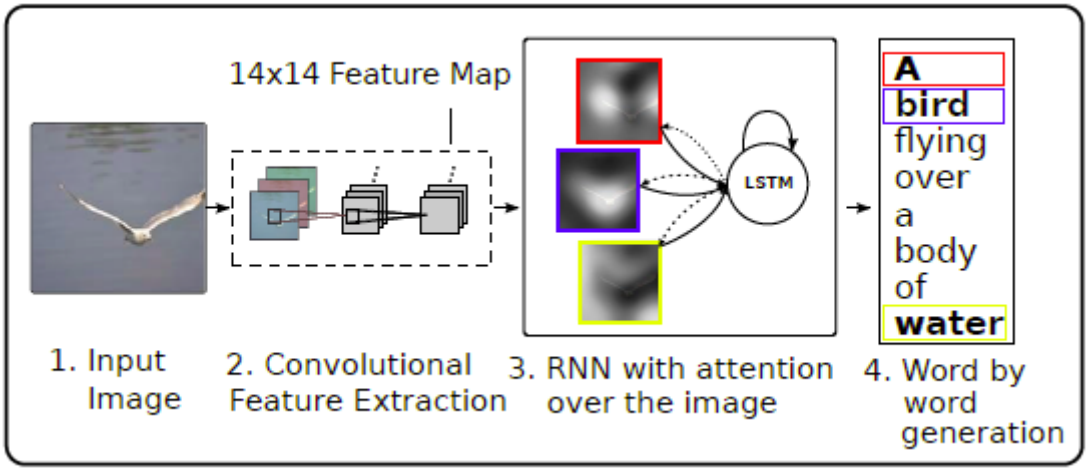
image caption是CV的最初始目标，不仅要获得图片里的物体，还要表达他们之间的关系。最近的研究都是基于NN的模型架构，特别是RNN和sequence to sequence的应用，主要灵感来自机器翻译。目前，NN方法的主要研究分为以下几个方向：1、用单独的CNN来获取图像的特征，然后，利用这些特征进行生成句子（排序，检索，生成）；2、将CNN获取的特征和句子特征联合嵌入到一个空间内，然后从中进行选择最优描述；3、利用一些全新的机制，将CNN和RNN结合，目的在利用CNN的全局特征或者局部特征来指导描述的生成。本文就是第三种方法，利用attention机制将两者结合起来，且提出hard和soft两两种形式。

除了利用NN解决此问题，还有另外两种方法进行解决：1、使用模板的方法，填入一些图像中的物体；2、使用检索的方法，寻找相似描述。这两种方法都使用了一种泛化的手段，使得描述跟图片很接近，但又不那么准确。

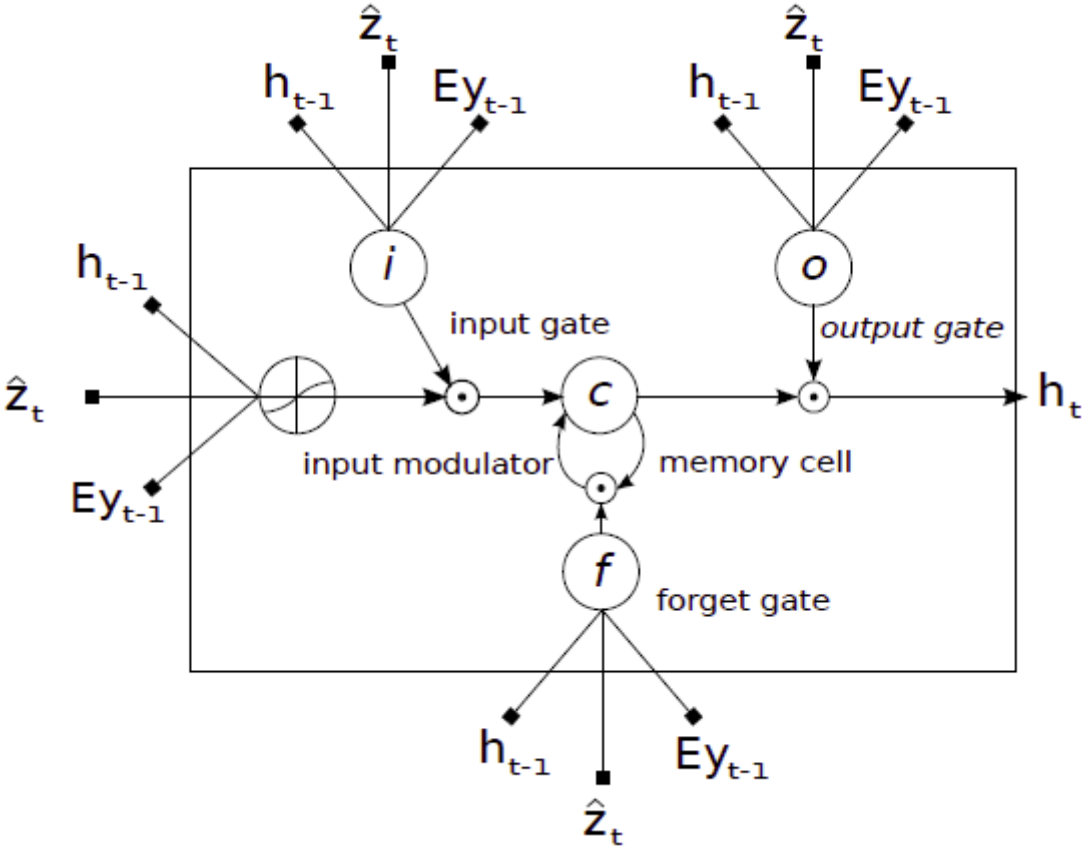
本文的贡献是1、提出两种attention机制利用在image caption任务中，hard和soft；2、利用可视化手段来清晰的理解attention机制的效果。

2、模型

模型分为两部分，如下图;



一部分编码部分encoder，目的是获取image的特征，不同于其他方法直接将最后全连接层的vector（反映图片整体特征）拿过来，此处，作者提取的是卷积层的输出，这样能够将局部的图片信息提取出来，分别进行生成sentence；另一部分为解码部分，采用LSTM模型，其结构如下：



具体的计算流程如下：

@马春杰杰你好： 想问下，你更改了源码没？可以输出每一类的recall,AP,以及mAP了吗？我也在做这步。...

--flysnow_88

5. Re:CV：object detection(YOLO)

@马春杰杰recall和mAP都是分类任务的指标，只是需要针对多标签任务进行一些修改，具体的，百度即可知道...

--AI-ML-DL

阅读排行榜

1. LSTM与GRU结构(8605)
2. 聚类算法（clustering）(3629)
3. CV：object recognition(ZFNet)(3615)
4. 生成对抗式网络(2707)
5. CV：image caption(Show, Attend and Tell: Neural Image Caption Generation with Visual Attention)(1700)

评论排行榜

1. CV：object detection(YOLO)(5)
2. 时间序列分析(1)
3. 聚类算法（clustering）(1)
4. 生成对抗式网络(1)

推荐排行榜

1. 时间序列分析(2)
2. CV：object recognition(ZFNet)(1)
3. LSTM与GRU结构(1)
4. 聚类算法（clustering）(1)

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

i、f、o、t、h是输入cell、遗忘cell、记忆cell、输出cell和隐藏层状态，Zt是文本向量，获取了相关信息，E是嵌入矩阵，zt是t时刻图像部分内容的动态表示，作者定义了一个机制，该机制产生一个正的权值α，该权值通过以下方式计算：

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})} \quad \hat{\mathbf{z}}_t = \phi(\{\mathbf{a}_i\}, \{\alpha_i\})$$

该权值表示，在已经产生的词序列的条件下，产生新词的时候，attention机制应该往哪儿看。其中，Φ是本文的关键函数。其中，隐藏层的初始状态通过两个分离的MLP实现

$$\mathbf{c}_0 = f_{\text{init,c}}(\frac{1}{L}\sum_{i}^{L}\mathbf{a}_i)$$

$$\mathbf{h}_0 = f_{\text{init,h}}(\frac{1}{L}\sum_{i}^{L}\mathbf{a}_i)$$

最后，通过以下式子来实现最后的输出：

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t))$$

此外，本文还讨论了两种attention机制：hard（随机）和soft（确定）。用st表示在生成第t个词时，模型决定attention的位置。把st看成一个隐变量，将zt看成随机变量，此时模型为关于α的多点分布，如下所示:

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i}$$

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i}\mathbf{a}_i.$$

还定义了一个目标函数，在提供图像的条件下，观测一系列词语的边界对数似然函数的下界。

$$\begin{aligned} L_s &= \sum_s p(s \mid \mathbf{a}) \log p(\mathbf{y} \mid s, \mathbf{a}) \\ &\leq \log \sum_s p(s \mid \mathbf{a}) p(\mathbf{y} \mid s, \mathbf{a}) \\ &= \log p(\mathbf{y} \mid \mathbf{a}) \end{aligned}$$

$$\frac{\partial L_s}{\partial W} = \sum_s p(s \mid \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} \mid s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid s, \mathbf{a}) \frac{\partial \log p(s \mid \mathbf{a})}{\partial W} \right]$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) \frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} \right]$$

学习算法的参数可以通过对损失函数直接求导得来，具体实现，可通过蒙特卡洛抽样得到关于模型参数的偏导。为了减小估计方差，有多种方案，详见论文，本文采用的是如下形式：

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \lambda_r(\log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) - b)\frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} + \lambda_e \frac{\partial H[\tilde{s}^n]}{\partial W}\right]$$

其中有两种超参数，由交叉验证得到，原理类似增强学习。此方法在已知参数α的分布前提下，在每个时间每个点返回一个由模型函数Φ确定的a，所以是一种比较hard的模式。

相比之下，soft attention放弃了hard attention中每一时间都要抽样一个attention区域的方式，而是直接产生，如下所示：

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L}\alpha_{t,i}\mathbf{a}_i$$

该形式使得整个模型平滑可区分，且通过使用向后传播使端到端的学习不那么重要。我们定义如下式子：

$$NWGM[p(y_t = k \mid \mathbf{a})] = \frac{\prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i \exp(n_{t,j,i})^{p(s_{t,i}=1|a)}} = \frac{\exp(\mathbb{E}_{p(s_t|a)}[n_{t,k}])}{\sum_j \exp(\mathbb{E}_{p(s_t|a)}[n_{t,j}])}$$

该式子表示caption预测的归一化加权几何平均可以通过文本向量近似得很好。也表示softmax的归一化加权几何平均可以通过把softmax应用到潜在线性预测的期望中获得。简单来说，确定的attention是关于attention位置的边缘似然的近似。

在训练确定attention时，我们引入了一个双向随机正则，目的是为了让attention平均的对待图片的每一部分。此外，soft attention还有一个阈值β，如下所示：$\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$，系统最后就是最小化以下带有惩罚项的负对数似然函数：

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C}\alpha_{ti})^2$$

3、实验

数据集采用Flickr8k、Flickr30k和MS COCO。评价准则使用BLUE和METEOR。具体实验过程，以及使用的参数，详见论文内容。以下是一些实验结果：

		BL	
Dataset	Model	B-1	B-2
Flickr8k	Google NIC(Vinyals et al., 2014)†Σ	63	41
	Log Bilinear (Kiros et al., 2014a)°	65.6	42.4
	Soft-Attention	**67**	44.8
	Hard-Attention	**67**	**45.7**
Flickr30k	Google NIC†°Σ	66.3	42.3
	Log Bilinear	60.0	38
	Soft-Attention	66.7	43.4
	Hard-Attention	**66.9**	**43.9**
COCO	CMU/MS Research (Chen & Zitnick, 2014)a	—	—
	MS Research (Fang et al., 2014)†a	—	—
	BRNN (Karpathy & Li, 2014)°	64.2	45.1
	Google NIC†°Σ	66.6	46.1
	Log Bilinear°	70.8	48.9
	Soft-Attention	70.7	49.2
	Hard-Attention	**71.8**	**50.4**

4、总结

可知，attention的加入，能够显著提高描述的性能，并且可分为hard和soft两种attention机制，hard更难进行训练和理解，但hard相对soft，其提高并没有很明显，需要继续改进和提高。

分类: CV

AI-ML-DL
关注 - 0
粉丝 - 15
+加关注

0 0

« 上一篇：KNN（k-Nearest Neighbor）算法
» 下一篇：朴素贝叶斯算法（Naive Bayesian）

posted @ 2016-11-29 14:19 AI-ML-DL 阅读(1700) 评论(0) 编辑 收藏