show attentd and tell 模型理解

> attention机制就是为了实现在生成一个词时去关注当前所"应该"关注的显著（salient）信息这一目的，手段就是**对输入信息的各个局部赋予权重**。

论文的caption模型中：

1.在解码的每个时刻都会接收由attention机制所计算出的编码向量。

2.使用低层卷基层的张量作为CNN的输出。

3.两种attention（soft和hard）通过 ϕ 函数来控制。

> NIC模型使用CNN最后一层输出向量表示作为图像特征的缺点是丢失了能够使caption变得更丰富的一些信息。
>
> NIC那篇论文提到，我们直接下载的的预训练CNN模型由于是在分类数据集训练的，默认抛弃了诸如颜色等对分类没有帮助的特征，这就已经造成了图像信息的部分丢失。

1. 模型结构

1.1 encoder：卷积特征

在encoder端，模型使用CNN来提取 L 个 D 维张量，每一个都对应图像的一个区域：

$a = a_1, \ldots, a_L, a_i \in R^D$

> 这里的张量就是CNN低层卷基层的张量输出，使得decoder可以通过选择所有特征向量的子集来选择性地聚焦于图像的某些部分。
>
> ？？？为什么低层的卷基层输出代表的是图像的一部分？不应该是代表着依次学习到的轮廓，颜色，材质之类的信息吗，更高层的才有图像的更精确的信息。所以这个低层和高层该怎么划分？

1.2 decoder：LSTM

1.2.1 LSTM的输入

抛去隐状态和词向量输入不谈，这里真正的富含图像信息的输入是捕捉了特定区域视觉信息的上下文向量（context vector）

（1）它和时刻 t 有关，是一个动态变化的量，在不同的时刻将会捕捉到与本时刻相对应的相关图像区域。

（2）这个量将由attention机制计算得到，而且在每一时刻都输入decoder。

1.2.2 隐状态和细胞初始状态

隐状态和细胞状态的初始值的计算方式：使用两个独立的多层感知机，感知机的输入是各个图像区域特征的平均：

$c_0 = f_{init,c}(\frac{1}{L}\sum_{i=1}^{L} a_i)$

$h_0 = f_{init,h}(\frac{1}{L}\sum_{i=1}^{L} a_i)$

基于隐状态，就可以计算词表中各个词的概率值。

取概率最大的那个作为当前时刻生成的词，并将作为下一时刻的输入。

其实就是个全连接层：

$p(y_t|a, y_1, \ldots, y_{t-1}) \propto exp(L_o(Ey_{t-1} + L_h h_t + L_z z_t))$

2.attention 机制

> 通过attention机制计算出的 z_t 被称为 context vector，是捕捉了特定区域视觉信息的上下文向量。

(1)目的：在解码的不同时刻可以关注不同的图像区域，生成更合理的词。

因此关键的量为：解码时刻t，输入序列的区域a_i

(2)实现方式：在时刻 t，为输入序列的各个区域 i 计算出一个权重α_{ti} 。

因为需要满足输入序列的各个区域的权重是加和为一的，所以使用Softmax来实现这一点。

(3)Softmax需要输入的信息：被计算的区域 a_i ;上一时刻 t-1 的信息 h_{t-1} ,计算公式如下：

$e_{ti} = f_{att}(a_i, h_{t-1})$

$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})}$

式中的 f_{att}是耦合计算区域 i 和时刻 t 这两个信息的打分函数。文中使用多层感知机。 现在，有了权重，就可以计算 z_t了：

$z_t = \phi(a_i, \alpha_{ti})$

（其中函数 ϕ代指文中提出的两种attention机制，对应于将权重施加到图像区域到两种不同的策略。）

2.1 hard attention

权重 α_{ti}是图像区域 a_i 在时刻 t 被选中作为输入decoder的信息的概率，有且仅有一个区域会被选中。为此，引入变量 $s_{t,i}$ ，当区域 i 被选中时取值为 1 ，否则为 0 。

$z_t = \sum_i s_{t,i} a_i$

如何求$s_{t,i}$：

在该篇论文中，将s_t 视作隐变量，为参数是 $\{\alpha_i\}$ 的多元伯努利分布：

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

后面利用类似EM算法的思路，利用jensen不等式转化出目标函数Ls

$logp(y|a) = log\sum_s p(s|a)p(y|s,a) \geq \sum_s p(s|a)logp(y|s,a)$

$L_s = \sum_s p(s|a)logp(y|s,a)$

式中的 y 是为图像 a 生成的一句caption（由一系列one-hot编码构成的词序列）

为了能够使反向传播过程中目标函数可微，使用蒙特卡洛方法来近似目标函数的梯度。

$$\tilde{s}_t \sim \text{Multinoulli}_L(\{\alpha_i\})$$

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

关于em算法与jensen不等式，参见：

1.感性的理解EM算法： https://www.jianshu.com/p/1121509ac1dc

2.EM算法的推导：https://www.cnblogs.com/pinard/p/6912636.html

关于蒙特卡洛方法，参见：

https://blog.csdn.net/pH646463981/article/details/80715736

2.2 soft attention

权重 αti 是图像区域 ai 在时刻 t 的输入decoder的信息中的所占的比例。

将各区域 ai与对应的权重 αt,i做加权求和就可以得到 z_t：

$E_{p(st|a)}[z_t] = \sum_{i=1}^{L} \alpha_{t,i} a_i$

这个模型是光滑的，可以使用BP算法通过梯度进行学习。文章定义了归一化加权几何平均值（NWGM）

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

该式表示caption的结果可以通过文本向量很好近似。

也表示soft attention是关于attention位置的边缘似然的近似。

在训练soft attention时，文章引入了一个双向随机正则，目的是为了让attention平等的对待图片的每一区域。

另外，还定义了阈值β，目的是让解码器决定是把重点放在语言建模还是在每个时间步骤的上下文中。

$$\phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \beta \sum_i^L \alpha_i \mathbf{a}_i$$

$$\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$$

Soft attention最终是通过最小化下式进行训练：

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