**一.主要思想**

1，本文使用visual-semantic embedding的方法来计算reward，用图片和图片的正负描述句子，通过最小化损失函数的方法，学习得到图片和句子的余弦相似度作为reward，记为vp（s）；

2，引入value网络vtheta（s），通过最小化平方误差函数的方法使其逼近vp（s）；

3，policy网络使用强化学习方法，以vtheta（s）为baseline，进行训练；

4，由于policy网络的action是词的one-hot向量，one-hot向量的维度特别大训练困难，

故提出一种训练方法的trick，即对一个长度为T的句子，每次固定前T-i\*delta个词，用强化学习的方法训练后i\*delta个词，以此来更新RNN网络，其中i=1,2，直到训练整个完整的句子，以此来降低网络参数收敛的难度；

5，本文在测试阶段使用Beam Search方法，每次选择前k个最可能的输出继续下一步的训练，

但不同于一般以policy的输出概率作为选择最可能输出的词的标准，本文还在policy输出概率的基础上+vtheta作为新的选择标准，

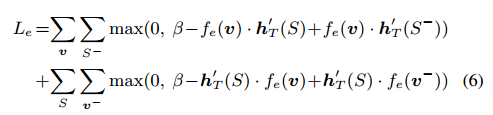
这种方法叫做lookahead inference，即不光考虑了policy输出概率的local guidance，还考虑了value值的global guidance；

**二.细节**

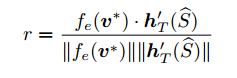
1，vp（s）的计算

首先给出一张图和这张图的正负描述句子，用CNN卷积图片得到图片特征，再用f（）对图片特征进行embedding，同时把句子输入RNN，用the last hidden state of RNN，即h'T（S）作为句子特征，

固定CNN的参数，通过最小化损失函数来学习RNN参数和f（）参数，损失函数如下：



学习到网络参数后，我们把一张图和一个句子输入网络，进行如下计算得到图片和句子的余弦相似度作为reward

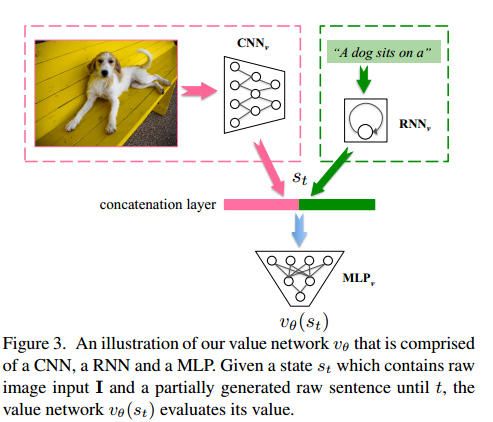


2，vtheta（s）结构

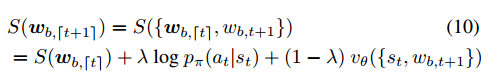
注意vtheta（s）可对每一个state（训练得到不完整句子的输出）做一次计算，vp（s）是对一个完整sentense输出后的state做计算；

换言之，每输出一个完整的句子得到一个reward，每输出一个词得到一个vtheta，

故训练时，我们采用当前句子的reward和当前句子中随机采样的state对应的vtheta，进行一次梯度下降更新参数；



3，lookahead inference的计算公式



**三、idea/改进方向**

1、强化学习reward的计算方式；

2、强化学习value网络的网络结构；

3、Beam Search选择最可能输出的标准；