**Compositional Vector Grammars**

CVG模型主要是用来将一个句子映射到一个与词相同的向量空间的模型。CVG是一个结合了PCFG（概率上下文无关）和RNN（递归网络）各自优点的模型，PCFG能够将短语划分到某一离散的类别（如NP，VP），RNN能够捕获到短语的详细信息，某些具有二义性的语法可以通过语义来消除二义性，如They ate udon with forks. vs. They ate udon with chicken。也就是说，CVG能够同时捕获短语的语法结构和不同结构所表达的信息。

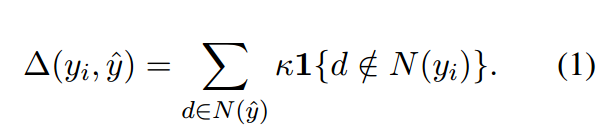
**Word Vector Representations**

此处的词向量采用Collobert和Weston提出的嵌入式模型（详见：lec5），该模型返回一个look up table，记为X，假设现在现在有一个包含m个词的句子S，每个单词w对应于X中的索引为[w] = i，对应的词向量为,我们将句子S作为一个（词，词向量）序列，

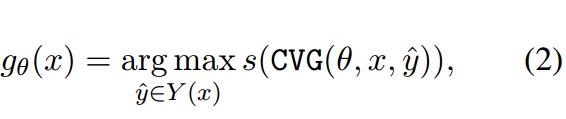
**Max-Margin Training Objective for CVGs**

我们期望能学习到这样一个函数，其中是句子序列集合，是所用可能的带标签的二叉解析树的集合，对于给定的句子，所有可能的二叉解析树集合用表示，真实二叉解析树记为，预测得到二叉解析树记为

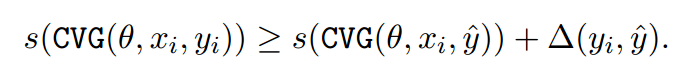
定义为预测为的损失值，当这颗预测树错的越厉害时，它的损失值越大，两棵树的差值应当被认为是预测树与真实树之间不同的节点的个数，树y的节点集合记为*N*(*y*)，则损失函数由下式计算所得：



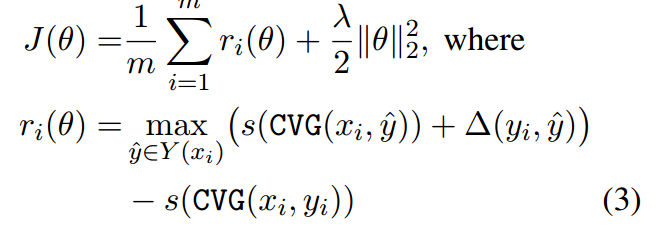
*κ* 是一个比例系数，原文作者在实验时将其设置为了0.1，我们的目标是搜索在参数θ下得分（用s函数计算）最高的树，也就是使得下式成立的那颗树。



Max-margin结构预测目标在训练CVG时，希望正确类别对应的树的得分最高，且这个得分至少要比训练得到的书大一个损失值

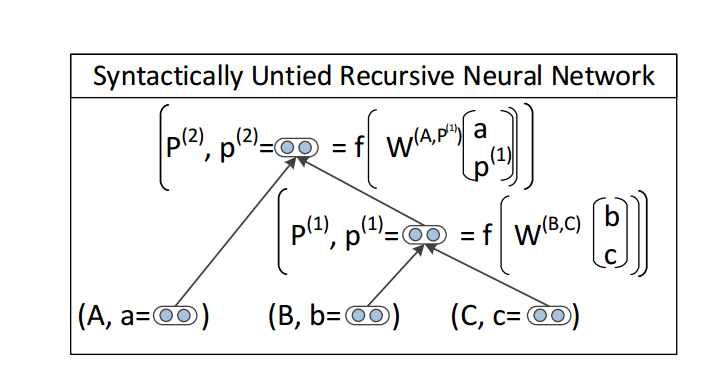


为此，对于m个训练样本，设定如下目标函数

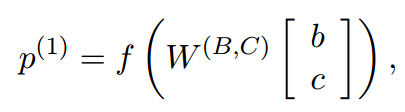


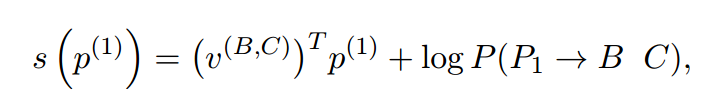
**Scoring Trees with CVGs**

下图显示了一个SU-RNN的结构，与传统的RNN网络相比较，SU-RNN对于合并函数中所使用的权值矩阵W放宽的限制，在传统RNN中，不论待合并的俩个子短语是什么类型（这里的类型是指NP，VP等），都使用同一个W。而SU-RNN中，事先通过PCFG捕获短语的类别，每一个节点包含两个信息，短语类别和短语向量（分别对应着下图中的A，a，注意，A不是一个词矩阵，这与后一章的模型不同）。因此，在合并时权重矩阵就依靠子节点的两个类别信息进行选择。因此，SU-RNN结合了离散的、概率语法规则和连续的向量的信息。直观上，一个NP应该和它的头名次相似，并且很少受到另一个决定因素的影响，而在一个形容词中，两个词共同决定了形容词意思。SU-RNN显然能更好的适应这一层意思。



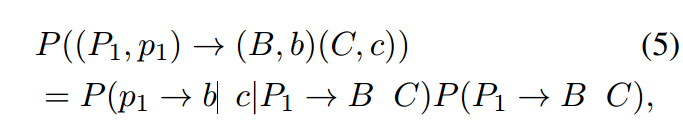
原始的RNN仅包含一个权重矩阵W，而CVG模型中，权重矩阵的规模取决于词性类别的个数，以上图为例，的计算过程如下：



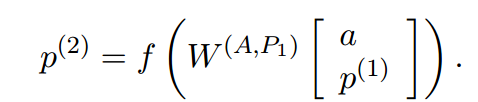


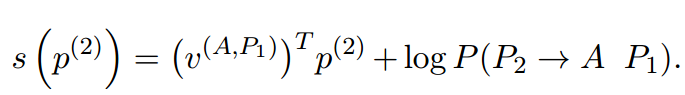
式子中各个向量的维度如下：

S是得分函数，P(P1->BC)是该生成式的概率,该信息由PCFG获得。得分函数可以被看作下述公式



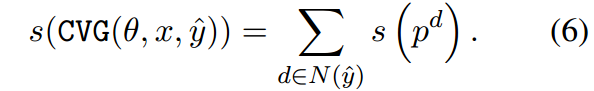
需要注意的是，由于词向量的连续性，基于CVG规则得分s与（5）没有可比性，因为后者对于所有的子节点的和的值为1，p2的计算公式如下





**Parsing with CVGs**

上面介绍了一个节点的得分信息，一颗树的得分就是所有节点得分的和，通过下式计算：



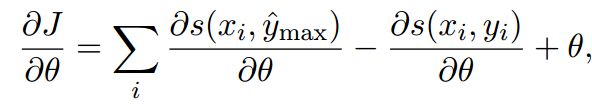
目标函数（3）中包含一项，对于一个长句子来说，找到句子中词的所有排列几乎是不可能的。基于各种原因，作者们最后决定使用俩个自底向上的解析表，第一个解析表仅使用采用CKY算法的PCFG模型。这一步，使得解析树的根节点保存top k个最优解析，这里采用(Huang and Chiang, 2005)Better k-best parsing. 算法。第二个解析图中，采用完全的CVG模型，（采用所有可能的权重举证），这里CVG模型计算树的得分时只使用了上述PCFG或得的top k个树。还需要注意的一点就是，一个节点的得分只通过它的两个字节点计算，而不是他的子孙节点。

**Training SU-RNNs**

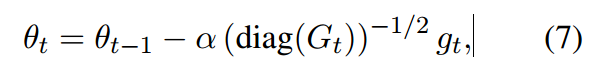
训练分两步进行，第一步先训练PCFG模型，并且得到top k个树，之后用这top k个树训练CVG，CVG采用等式（3）训练。在训练第i个句子时，需要更新所有出现过在这棵树上出现过的。

**Subgradient Methods and AdaGrad**

另表示所有的参数集合，其中，表示所有权重矩阵的集合。等式（3）的子梯度公式变为：

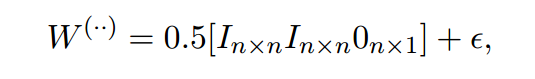


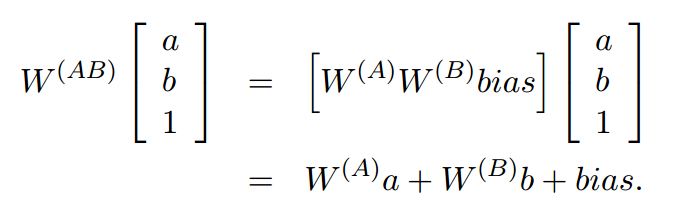
学习率采用AdaGrad，首先定义是第tao次的子梯度，定义，则梯度更新为



采用自适应学习率算法是应为有些矩阵W可能出现的次数比较少，因此一开始应该给较大的学习率，防止欠学习。

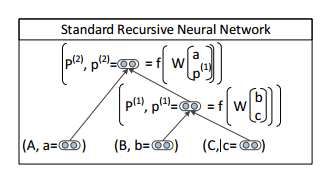
**Initialization of Weight Matrices**





**Recursive Neural Network**

一个典型的递归神经网络具有如下结构



各个节点的计算公式如下：

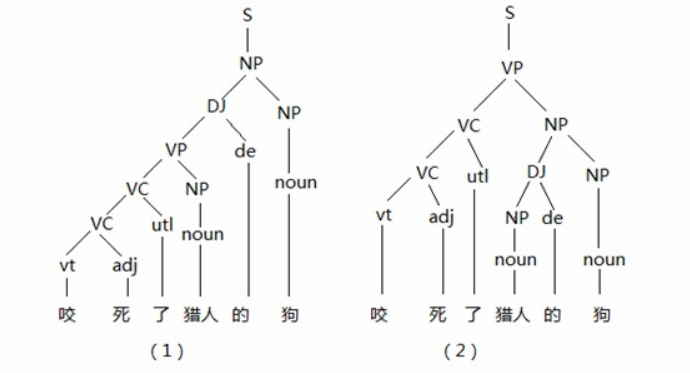




在标准RNN中，W只有一个。

**Probabilistic Context Free Grammar**

PCFG主要是用来构建具有二义性文法的语法树的算法，在使用CFG模型对某个具有歧义的句子构造语法树时，往往可以构建出多棵语法树，比如



因此，提出了基于概率的上下文无关文法PCFG，先叙述一下PCFG的定义

一个概率上下文无关文法（PCFG）是一个五元组(N,∑,S,R,P)：

（1）一个非终结符集N

（2）一个终结符集∑

（3）一个开始非终结符S∈N

（4）一个产生式集R

（5）对于任意产生式r∈R，其概率为P(r)

PCFG是[CFG](https://baike.baidu.com/item/CFG)的扩展，PCFG的规则表示形式为：A→α p，其中A为非终结符，p为A推导出α的概率，即p=P(A→α)，该概率分布必须满足如下条件：

∑P(A→α)=1

也就是说，相同左部的产生式概率分布满足归一化条件。分析树的概率等于所有使用规则概率之积。

当构建出多棵语法树时，我们就选取树中所有节点概率之积最大的树。这就是PCFG的总体思想，更详细的算法可以参考以下博客

http://blog.csdn.net/mingspy/article/details/8843964