NLP（主要来自cs224总结（2018））

问题：

* Dependency Parsing（lecture5）后面有需要再想看，没怎么看懂
* 训练终止条件是什么

1. 自己总结

* 注意结合程序体会文中复杂公司，如词嵌入的公式
* 入门建议网页：香侬科技李纪为：初入NLP领域的一些小建议
* 自训练方法：遗珠之作？谷歌Quoc Le这篇NLP预训练模型论文值得一看
* Notes3有各激活函数的图，注意导数范围

1. 参数调节

* 对于tanh和sigmoid激活函数采用uniform distribution初始化比较好
* 优化过程中可使用滑动平滑方法更新

1. 超参数

* 学习率一般需要刚开始较大，后面较小。Note3最后提了一些方法。

1. NLP系统总结

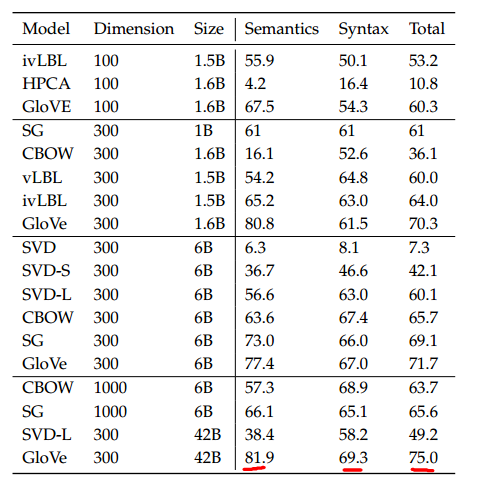
* 预训练

使用预训练的词向量可将离散表示（one-hot）转换成分布式表示，解决了稀疏、离散、高维、不易计算机处理（计算机善于解决数据连续数据计算），生成词向量常用CROW、skip-gram,使用n-gram方法在文本中移动，使用其隐层的向量作为词向量。如果任务的数据集足够大可以在任务中重新训练词向量提高模型表现，但数据集较小时不宜采用。

词向量表示方式：为了更好地完成大多数NLP任务，词向量应该能表现单词之间的相似性和差异性。通常，窗口大小越小，语法测试的性能越好;窗口越宽，语义测试的性能越好

* one-hot：不能表示词之间的相似性，两向量相乘为0
* SVD Based Methods：基于统计，
* Word-Document Matrix：假设相似的词更有可能在同一篇文章出现
* Window based Co-occurrence Matrix：两词在不同句出现的次数
* Applying SVD to the cooccurrence matrix：取特征值前k个向量，特征值大的表示比较重要的向量，这个有待论证
* CBOW: 基于n-gram窗口模型，由于相似矢量的点积更高，它将推动相似的词彼此接近以获得高分。相似词上下文向量相似，则v相似，后面为了使对应词概率都为1，两个位置的向量对应向量会比较接近，画出矩阵更易理解
* Skip-grams：基于n-gram窗口模型，
* Glove（global vector for word representation）：基于全局统计，利用skip-gram的方法，代表最高水平

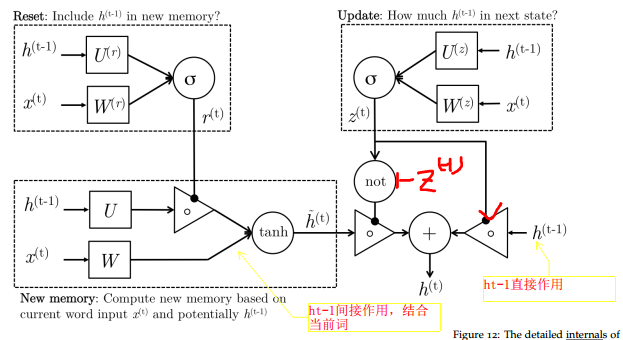
下图是评价结果图，发现词向量表现和模型（glove最好，窗口大小8最佳）、数据集大小（越大越好）、维数（越高越好）、窗口大小（越大越好）都有关

* 预处理

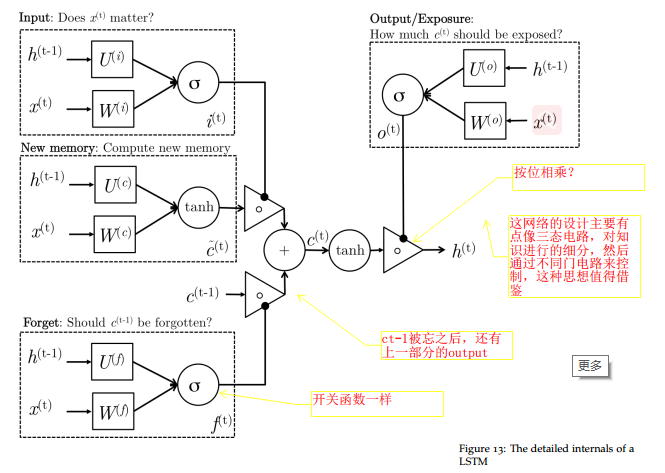
需要将文本转换成词向量，并使用填充等统一句子长度，使用特定开始、结束、填充标记，并用乘mask（0）来消除其对结果影响；需要将数据归一化、中心化，或使用SVD使其具有指定方差

分词：英文自然用空格分开，不同形态（时态等）看成不同词，中文分词较复杂

* 常用模型
* RNN:梯度消失或梯度爆炸。隐层输出为一个向量，输出是所有可能词汇的one-hot表示，可以将每个时间步输出和语言模型P（wi|w1…wi-1）对应
* 优点：可以输入不同长度、对不同长度模型不变、理论上可以利用前面的所有信息、各个时间步之间参数共享
* 缺点：梯度爆炸和消失，串行传播、计算较慢
* GRU：



* LSTM：<https://zybuluo.com/hanbingtao/note/581764>



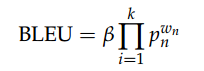
* seq2seq：翻译、问答、总结等问题，先编码完再译码的优点：可以调整输出语序、对于每个输出关注所有输入的不同。注意seq2seq中输入已经全部知道，要根据不同任务分析输入输出的情况，单向时逆序可能提高翻译效果，第一个词先翻译了。双向是为了捕获两个方向的关系

结果的搜索方法：

* Exhaustive search：全部列出来，选最好的
* Ancestral sampling：条件概率
* Greedy Search：每步最大概率
* Beam search：每次选K个

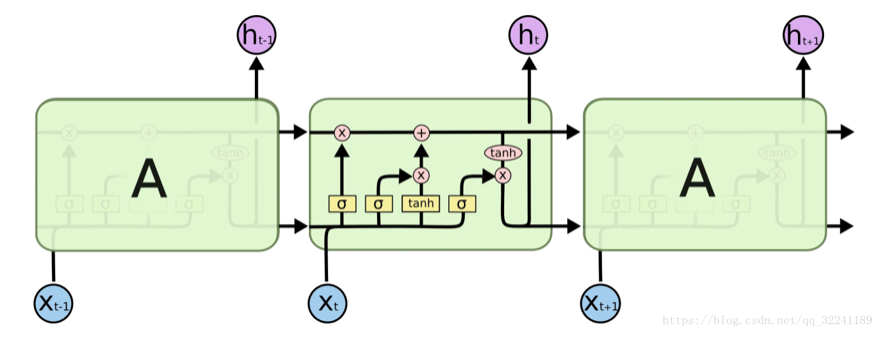
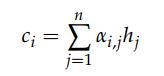
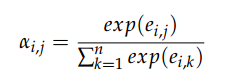
评价方法

* 人工
* 用另一个任务
* BLEU：不许呀要对应位置，k取4左右



处理输出词汇太多问题

* 限制softmax大小：Noise Contrastive Estimation（负采样）和树结构（Hierarchical Softmax，log（n））
* 标记<UNK>具体方法见notes6
* attention：目的是关注输入的各个部分，具体关注公式不确定，根据公式的不同有各种各样不同的attention机制，对于seq2seq问题，主要是建立decoder隐层s与各个encoder的隐层h之间的关系，如下主要是a函数的不同。其能翻译更长句子。



Ct

Ct-1

ht-1

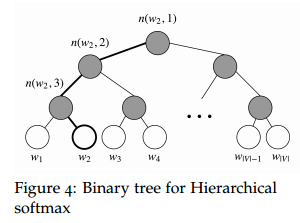
ht

* CNN：其也可用于处理序列，不那么常用
* 递归神经网络（Recursive Neural Networks）：树结构，有利于运用句法（主谓宾之类的），第一，理解词的语义，第二，分析句子结构（语法），第三利用前两者理解句子。短语采用词向量串联的形式。对不同类别的输入使用不同的W比对所有类别只使用一个W更自然。
* 1

1. 词嵌入：自然语言处理（NLP）中语言模型与表征学习技术的统称。概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量。词嵌入的方法包括人工神经网络、对词语同现矩阵降维、概率模型以及单词所在上下文的显式表示等。在底层输入中，使用词嵌入来表示词组的方法极大提升了NLP中语法分析器和文本情感分析等的效果。

猜想：以符号表示为输入，目标词为输出，训练权重，将权重矩阵看成词向量，只需句子训练就可。训练时把权重打印出来看看

1. 分级softmax（如下图，计算复杂度log|V|）相对普通softmax是一个更有效的选择。在实际应用中，分级softmax对于不频繁的单词比较好，而负抽样对于频繁的单词和低维向量比较好



1. 识别自然语言的几个难点：语言的歧义及含义模糊，有些意思模棱两可；人为了简化，可能省略某些词，如代词；语言主要用于交流，说话方式不规则，主要是为了人能听懂。
2. wordnet：nltk
3. 可以通过小批量梯度下降（32,64）减小噪声的影响，同时又不至于计算量过大
4. 正负样本：https://baike.baidu.com/item/负采样/22884020?fr=aladdin
5. 对于RNN输入句子的长度要求是有限的，也就是时间步是有限的，一般会展开使其更形象的表示出来
6. N-grams：<https://www.cnblogs.com/cdsj/p/5720391.html>；为什么含有大量0值？
7. CBOW和skip-grams，onehot，word2vec，具体连接方法？

<https://www.jianshu.com/p/471d9bfbd72f>

<https://www.cnblogs.com/pinard/p/7160330.html>

1. Attention模型

<https://blog.csdn.net/hahajinbu/article/details/81940355>

<https://blog.csdn.net/u014665013/article/details/82619808>

1. Memory-based neural networks（待看）

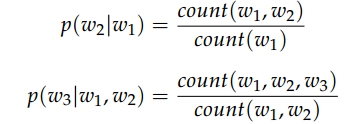
<https://blog.csdn.net/u014248127/article/details/84894739>

1. GRU

<https://blog.csdn.net/qq_28743951/article/details/78974058>

<https://www.cnblogs.com/jiangxinyang/p/9376021.html>

1. 语言模型：常用公式

n-gram Language Models：存在稀疏和占用内存大的问题

1. 1